

Grammatiken als formale Objekte

Gereon Müller

Institut für Linguistik
Universität Leipzig

SoSe 2006

www.uni-leipzig.de/~muellerg

Lit.:

Partee et al. (1993)

(1) Grammatik:

Eine Grammatik ist ein Quadrupel $\langle V_T, V_N, S, R \rangle$, wobei gilt:

- V_T = Vokabular (Alphabet) der terminalen Symbole;
- V_N = Vokabular (Alphabet) der nicht-terminalen Symbole (V_T und V_N sind disjunkt);
- S = Startsymbol;
- R = endliche Menge von Regeln der Form $\psi \rightarrow \omega$, wobei ψ und ω Ketten sind.

Interpretation der Regeln: Wenn ψ irgendwo als Teilkette auftritt, kann es durch ω ersetzt werden und so eine neue Kette erzeugen.

(2) Sprache (Chomsky (1957, 13)):

Eine Sprache ist eine (potentiell infinite) Menge von Ketten von terminalen Symbolen (= Sätzen), die durch eine Grammatik erzeugt werden.

Erzeugung einer Sprache

(3) Beispielgrammatik:

- a. $V_T = \{a,b\}$
- b. $V_N = \{S,A,B\}$
- c. $S \in V_N$

$$d. R = \left\{ \begin{array}{l} S \rightarrow ABS \\ S \rightarrow e \\ AB \rightarrow BA \\ BA \rightarrow AB \\ A \rightarrow a \\ B \rightarrow b \end{array} \right\}$$

Notationskonvention: Kleibuchstaben: terminales Alphabet; Großbuchstaben: nicht-terminales Alphabet.

(4) Erzeugung von 'abba':

- a. $S \Rightarrow ABS$
- b. $ABS \Rightarrow ABABS$
- c. $ABABS \Rightarrow ABAB$
- d. $ABAB \Rightarrow ABBA$
- e. $ABBA \Rightarrow ABbA$
- f. $ABbA \Rightarrow aBbA$
- g. $aBbA \Rightarrow abbA$
- h. $abbA \Rightarrow abba$

Diese Grammatik erzeugt die Sprache L_0 .

(5) $L_0: \{x \in \{a,b\}^* \mid x \text{ enthält die gleiche Anzahl von a's und b's}\}$

Kleene-Stern:

A^* bezeichnet die Menge aller Ketten, die über dem Alphabet A gebildet werden können (der Abschluss oder 'Kleene-Stern' auf einer Menge von Ketten).

- **Rekursion** (Chomsky (1957))

- **Rekursion** (Chomsky (1957))
Hierarchische Strukturierung komplexer sprachlicher Ausdrücke mit der Möglichkeit wiederkehrender Muster innerhalb der Struktur.

Kerneigenschaften von Grammatiken natürlicher Sprachen

- **Rekursion** (Chomsky (1957))
Hierarchische Strukturierung komplexer sprachlicher Ausdrücke mit der Möglichkeit wiederkehrender Muster innerhalb der Struktur.
- **Doppelte Artikulation** (Martinet (1960))

Kerneigenschaften von Grammatiken natürlicher Sprachen

- **Rekursion** (Chomsky (1957))
Hierarchische Strukturierung komplexer sprachlicher Ausdrücke mit der Möglichkeit wiederkehrender Muster innerhalb der Struktur.
- **Doppelte Artikulation** (Martinet (1960))
Separierung sprachlicher Ausdrücke in (a) minimale Einheiten, die Bedeutung tragen (Morpheme), und (b) minimale Einheiten, die Bedeutung unterscheiden (Phoneme).

Grammatiken formaler Sprachen wie (3) können Rekursion aufweisen, aber sie haben keine doppelte Artikulation: Die terminalen Knoten a, b sind primitive Einheiten des Strukturaufbaus (Syntax/Morphologie; Wörter/Morpheme), die keine weiter zerlegbare interne Struktur haben (wie das bei Wörtern/Morphemen natürlicher Sprachen der Fall ist: **a** steht etwa für /t-a:-l/, **b** für /ʃ-a:-l/, **c** für /ʃ-t-a:-l/, usw.

Bäume:

Syntaktische Bäume erfüllen eine Reihe von Wohlgeformtheitsbedingungen.

(6) **Bedingung der singulären Wurzel:**

In einem wohlgeformten Strukturbaum gibt es genau einen Knoten, der jeden anderen Knoten dominiert.

(7) **Exklusivitätsbedingung:**

In einem wohlgeformten Strukturbaum gilt für alle Knoten x und y : x und y stehen in einer Präzedenzbeziehung P (d.h., entweder $\langle x,y \rangle \in P$, oder $\langle y,x \rangle \in P$) gdw. x und y nicht in einer Dominanzrelation D stehen (d.h., es gilt weder $\langle x,y \rangle \in D$, noch $\langle y,x \rangle \in D$).

(8) **Bedingung der Nicht-Verwirrung:**

In einem wohlgeformten Strukturbaum gilt für alle Knoten x und y : Wenn x y vorangeht, dann gehen alle Knoten, die von x dominiert werden, allen Knoten, die von y dominiert werden, voran.

Von Grammatiken zu Bäumen

Die durch grammatische Regeln erzeugten Ketten korrespondieren Strukturbäumen.

- (9) Eine Grammatik (mit Regeln der Art $A \rightarrow \psi$) erzeugt einen Baum gdw. wenn gilt:
- Die Wurzel ist mit dem Startsymbol der Grammatik gelabelt.
 - Die Kette, die durch die gemäß der Präzedenzrelation geordneten Blätter des Baumes gebildet wird, ist eine Kette von terminalen Symbolen der Grammatik.
 - Für jeden Teilbaum der Form $[A \alpha_1 \dots \alpha_n]$ im Baum (wobei $A \alpha_1 \dots \alpha_n$ unmittelbar dominiert) gibt es eine Regel $A \rightarrow \alpha_1 \dots \alpha_n$ in der Grammatik.

Beispiel

(10) Beispielgrammatik:

a. $V_T = \{a,b\}$

b. $V_N = \{S,A,B\}$

c. $S \in V_N$

d. $R = \left\{ \begin{array}{l} S \rightarrow AB \\ A \rightarrow aAb \\ A \rightarrow e \\ B \rightarrow Bb \\ B \rightarrow b \end{array} \right\}$

(11) Erzeugte Bäume:

a. $[S [A a [A e] b] [B [B b] b]]$

b. $[S [A e] [B b]]$

c. $[S [A e] [B [B b] b]]$

d. $[S [A a [A e] b] [B b]]$

Die Chomsky-Hierarchie

Beobachtung:

Gemäß der Art der zugelassenen Regeln unterscheiden sich Grammatiken bezüglich ihrer **generativen Kapazität** (Mächtigkeit).

(12) Beschränkungen für Regeln:

a. **Typ-0-Grammatiken:**

–

b. **Typ-1-Grammatiken:**

Jede Regel hat die Form $\alpha A \beta \rightarrow \alpha \psi \beta$, wobei $\psi \neq \epsilon$.

c. **Typ-2-Grammatiken:**

Jede Regel hat die Form $A \rightarrow \psi$.

d. **Typ-3-Grammatiken:**

Jede Regel hat die Form $A \rightarrow xB$ oder $A \rightarrow x$.

Es gilt:

(i) α, β, ψ sind beliebige Ketten (u.U. leere) über der Vereinigung der terminalen und nicht-terminalen Alphabete.

(ii) A, B sind nicht-terminale Symbole.

(iii) x ist eine Kette von terminalen Symbolen.

Bemerkung:

- Typ-0-Grammatiken \leftrightarrow unbeschränkte Ersetzungssysteme

Bemerkung:

- Typ-0-Grammatiken \leftrightarrow unbeschränkte Ersetzungssysteme
- Typ-1-Grammatiken \leftrightarrow kontextsensitive Grammatiken

Bemerkung:

- Typ-0-Grammatiken ↔ unbeschränkte Ersetzungssysteme
- Typ-1-Grammatiken ↔ kontextsensitive Grammatiken
- Typ-2-Grammatiken ↔ kontextfreie Grammatiken

Bemerkung:

- Typ-0-Grammatiken \leftrightarrow unbeschränkte Ersetzungssysteme
- Typ-1-Grammatiken \leftrightarrow kontextsensitive Grammatiken
- Typ-2-Grammatiken \leftrightarrow kontextfreie Grammatiken
- Typ-3-Grammatiken \leftrightarrow reguläre Grammatiken (finite state grammars)

(13) Festlegung:

Eine Sprache ist vom Typ n gdw. wenn sie generiert wird von einer Grammatik vom Typ n .

Konsequenz:

Eine Sprache kann von mehr als einem Typ sein. L_0 z.B. kann durch eine Typ-0-Grammatik erzeugt werden (nämlich die in (3)); aber auch durch eine (kontextfreie) Typ-2-Grammatik.

L_0 durch Typ-0-Grammatik

(5) $L_0: \{x \in \{a,b\}^* \mid x \text{ enthält die gleiche Anzahl von a's und b's}\}$

(3) Typ-0-Grammatik:

a. $V_T = \{a,b\}$

b. $V_N = \{S,A,B\}$

c. $S \in V_N$

d. $R = \left\{ \begin{array}{l} S \rightarrow ABS \\ S \rightarrow e \\ AB \rightarrow BA \\ BA \rightarrow AB \\ A \rightarrow a \\ B \rightarrow b \end{array} \right\}$

L_0 durch Typ-2-Grammatik

(14) Typ-2-Grammatik:

a. $V_T = \{a,b\}$

b. $V_N = \{S,A,B\}$

c. $S \in V_N$

d. $R = \left\{ \begin{array}{l} S \rightarrow e \\ S \rightarrow aB \\ S \rightarrow bA \\ B \rightarrow b \\ B \rightarrow bS \\ A \rightarrow a \\ A \rightarrow aS \\ A \rightarrow bAA \\ B \rightarrow aBB \end{array} \right\}$

Pumping-Lemma

(15) Pumping-Lemma für reguläre Sprachen:

Wenn L eine infinite reguläre Sprache über dem Alphabet Σ ist, dann gibt es Ketten $x, y, z \in \Sigma^*$, so dass $y \neq \epsilon$ und $xy^n z \in L$, für alle $n \geq 0$.

Bemerkung (s.o.):

Σ^* bezeichnet die Menge aller Ketten, die über dem Alphabet Σ gebildet werden können (der Abschluss oder 'Kleene-Stern' auf einer Menge von Ketten).

Beobachtung:

Mit dem Pumping-Lemma kann man nachweisen, dass eine Sprache **nicht** regulär ist. (Technik: Modus tollens)

(16) Modus tollens:

- Wenn A, dann B.
- Nicht B.
- Es folgt: Nicht A.

L₁: Eine reguläre Sprache?

Frage:

Ist L₁ in (17) eine reguläre Sprache? Hier müssen alle Ketten aus **n** Symbolen **a** bestehen, denen **n** Symbole **b** folgen. Falls ja, dann muss es für jedes **n** ein **x**, **y**, **z** geben, so dass **xyⁿz** in L₁ ist.

$$(17) L_1 = \{a^n b^n \mid n \geq 0\}$$

(18) Drei mögliche Belegungen für **y**:

- y** = eine Anzahl von **a**'s, der eine Anzahl von **b**'s folgt.
- y** = eine Anzahl von **a**'s.
- y** = eine Anzahl von **b**'s.

L₁: Argumentation

(15) Pumping-Lemma für reguläre Sprachen:

Wenn L eine infinite reguläre Sprache über dem Alphabet Σ ist, dann gibt es Ketten $x, y, z \in \Sigma^*$, so dass $y \neq \epsilon$ und $xy^n z \in L$, für alle $n \geq 0$.

(17) $L_1 = \{a^n b^n \mid n \geq 0\}$

1. Fall:

(i) $xyz \rightarrow x = \epsilon, y = ab, z = \epsilon$

$\rightsquigarrow ab$

(ii) $xyyz \rightarrow x = \epsilon, y = ab, z = \epsilon$

$\rightsquigarrow *abab$

L₁: Argumentation

(15) Pumping-Lemma für reguläre Sprachen:

Wenn L eine infinite reguläre Sprache über dem Alphabet Σ ist, dann gibt es Ketten $x, y, z \in \Sigma^*$, so dass $y \neq \epsilon$ und $xy^n z \in L$, für alle $n \geq 0$.

(17) $L_1 = \{a^n b^n \mid n \geq 0\}$

1. Fall:

(i) $xyz \rightarrow x = \epsilon, y = ab, z = \epsilon$

$\rightsquigarrow ab$

(ii) $xyyz \rightarrow x = \epsilon, y = ab, z = \epsilon$

$\rightsquigarrow *abab$

2. Fall:

(i) $xyz \rightarrow x = \epsilon, y = a, z = b$

$\rightsquigarrow ab$

(ii) $xyyz \rightarrow x = \epsilon, y = aa, z = b$

$\rightsquigarrow *aab$

L₁: Argumentation

(15) Pumping-Lemma für reguläre Sprachen:

Wenn L eine infinite reguläre Sprache über dem Alphabet Σ ist, dann gibt es Ketten $x, y, z \in \Sigma^*$, so dass $y \neq \epsilon$ und $xy^n z \in L$, für alle $n \geq 0$.

(17) $L_1 = \{a^n b^n \mid n \geq 0\}$

1. Fall:

(i) $xyz \rightarrow x = \epsilon, y = ab, z = \epsilon$

$\rightsquigarrow ab$

(ii) $xyyz \rightarrow x = \epsilon, y = ab, z = \epsilon$

$\rightsquigarrow *abab$

2. Fall:

(i) $xyz \rightarrow x = \epsilon, y = a, z = b$

$\rightsquigarrow ab$

(ii) $xyyz \rightarrow x = \epsilon, y = aa, z = b$

$\rightsquigarrow *aab$

3. Fall:

(i) $xyz \rightarrow x = a, y = b, z = \epsilon$

$\rightsquigarrow ab$

(ii) $xyyz \rightarrow x = a, y = bb, z = \epsilon$

$\rightsquigarrow *abb$

L₁: Keine reguläre Sprache!

Resultat:

L₁ ist keine reguläre Sprache, weil y nicht hochgepumpt werden kann (und die resultierende Kette dann immer noch Teil der Sprache ist) – entweder wird beim Hochpumpen die Reihenfolge von a,b problematisch, oder die relative Anzahl von a's und b's.

$$(17) L_1 = \{a^n b^n \mid n \geq 0\}$$

Sind natürliche Sprachen reguläre Sprachen?

Frage:

Ist Englisch (Deutsch, etc.) eine reguläre Sprache?

Beobachtung:

Der Schnitt einer regulären Sprache mit einer regulären Sprache liefert wieder eine reguläre Sprache.

Strategie:

Englisch wird mit einer bekannt regulären Sprache geschnitten; wenn das Pumping-Lemma die resultierende Sprache als nicht regulär erweist, ist bewiesen, dass Englisch nicht regulär ist.

Relativsätze

(19) Relativsätze im Englischen:

- a. The cat died.
- b. The cat the dog chased died.
- c. The cat the dog the rat bit chased died.
- d. The cat the dog the rat the elephant admired bit chased died.

(20) Form dieser Sätze:

(the + N)ⁿ (V_{trans})ⁿ⁻¹ V_{intrans}

- (21) a. A = {the cat, the dog, the rat, the elephant, ...}
b. B = {chased, bit, admired, ate, befriended, ...}

Bemerkung:

L₂ ergibt sich aus dem Schnitt von Englisch und der regulären Sprache L₃ = A*B*{died}

(22) L₂ = aⁿbⁿ⁻¹died | a ∈ A und b ∈ B

L_2 und Pumping-Lemma

Frage:

Was sagt das Pumping-Lemma zu L_2 ?

(15) Pumping-Lemma für reguläre Sprachen:

Wenn L eine infinite reguläre Sprache über dem Alphabet Σ ist, dann gibt es Ketten $x, y, z \in \Sigma^*$, so dass $y \neq \epsilon$ und $xy^n z \in L$, für alle $n \geq 0$.

(22) $L_2 = \{a^n b^{n-1} \mid a \in A \text{ und } b \in B\}$

Antwort:

Wie vorher lässt sich y nicht hochpumpen, ohne entweder die Abfolge oder die relative Anzahl zu zerstören, die von L_2 gefordert werden.

Konklusion:

L_2 ist nicht regulär, und damit auch nicht Englisch. Also müssen Grammatiken für natürliche Sprachen **mindestens kontextfrei** sein.

Allgemeine Konklusion

Stand der Dinge:

Genau dies ist die Annahme bei [Gazdar \(1981\)](#): Grammatiken sind kontextfrei, aber nicht mächtiger. (Grammatiken mit Transformationen des klassischen, 60er-Jahre-Typs sind unbeschränkte Ersetzungssysteme.)

Ausblick:

Seit den Achtzigerjahren weiß man, dass natürliche Sprachen stärkere Grammatiken als kontextfreie benötigen. Der Beweis rekurriert auf ein [Pumping-Lemma für kontextfreie Sprachen](#), und er basiert (u.a.) auf sog. 'cross-serial dependencies' im Schweizerdeutschen. Das Grundmuster ist aber ganz ähnlich: Mehrere verschränkte Abhängigkeiten setzen komplexere Grammatiken voraus.

Streitpunkt:

Es ist umstritten, ob generative Kapazität aus linguistischer (nicht: mathematischer) Perspektive überhaupt ein taugliches Mittel zur Bewertung von Grammatiken ist.

Phrasenstrukturgrammatiken vs. Verkettungsgrammatiken

(23) Kontextfreie Phrasenstrukturgrammatik:

- a. **Ersetzungsregeln 1:**
- (i) CP (=S) \rightarrow C TP
 - (ii) TP \rightarrow vP T
 - (iii) vP \rightarrow DP v'
 - (iv) v' \rightarrow VP v
 - (v) DP \rightarrow D NP
 - (vi) NP \rightarrow N
 - (vii) VP \rightarrow DP V
 - (viii) vP \rightarrow CP V
- b. **Ersetzungsregeln 2:**
- (i) V \rightarrow {glaubt, mag}
 - (ii) N \rightarrow {Studentin, Professorin}
 - (iii) D \rightarrow {die}
 - (iv) v \rightarrow { \emptyset }
 - (v) T \rightarrow { \emptyset }
 - (vi) C \rightarrow {dass}
- c. **Vokabular:**
 V_T : {glaubt, kennt, Studentin, Professorin, die, \emptyset , dass}

(24) Verkettungsgrammatik:

- a. **Verkettungsregel**
Für $\alpha: \{[*\Gamma^*](\succ[*\Delta^*]\succ\dots), \dots\}$, $\beta: \{[\Gamma], \dots\}$: Verkettung $(\alpha, \beta) \rightsquigarrow$
 $[\alpha \alpha: \{([*\Delta^*]\succ\dots), \dots\}, \beta]$
- b. **Linearisierungsregeln:**
- (i) [+V] < Komplement.
 - (ii) [-V] > Komplement.
 - (iii) [\pm V] > Spezifikator.
- c. **Vokabular:**
- (i) glaubt: {[V], [*C*]}
 - (ii) kennt: {[V], [*D*]}
 - (iii) Studentin: {[N]}
 - (iv) Professorin: {[N]}
 - (v) die: {[D], [*N*]}
 - (vi) \emptyset : {[v], [*V*] \succ [*D*]}
 - (vii) \emptyset : {[T], [*v*]}
 - (viii) dass: {[C], [*T*]}

- Chomsky, Noam (1957): *Syntactic Structures*. Mouton, The Hague and Paris.
- Gazdar, Gerald (1981): Unbounded Dependencies and Coordinate Structure, *Linguistic Inquiry* 12, 155–184.
- Martinet, André (1960): *Éléments de linguistique générale*. Armand Colin, Paris.
- Partee, Barbara, Alice ter Meulen & Robert Wall (1993): *Mathematical Methods in Linguistics*. Kluwer, Dordrecht.