Replicative Processes: Morphology

Master: 04-046-2011 (Morphologie: Flexion)
IGRA: 07, Topics in Morphology (seminar)

Tuesdays, 17:00–18:30, S102, NSG
SoSe 2015, Universität Leipzig

Institut für Linguistik
Gereon Müller
gereon.mueller@uni-leipzig.de
http://www.uni-leipzig.de/∼muellerg

Complexity of Replicative Processes: Culy (1985)

1. Chomsky-Hierarchie

(1) *Grammatik*:
Eine Grammatik ist ein Quadrupel <V_T,V_N,S,R>, wobei gilt:
a. V_T = Vokabular (Alphabet) der terminalen Symbole;
b. V_N = Vokabular (Alphabet) der nicht-terminalen Symbole (V_T und V_N sind disjunkt);
c. S = Startsymbol;
d. R = endliche Menge von Regeln der Form $\psi \rightarrow \omega$, wobei ψ und ω Ketten sind.

Interpretation der Regeln R: 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.

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

(3) Regeln der Beispielgrammatik:

$$\mathrm{R} = \left\{ \begin{array}{ccc} \mathrm{S} & \rightarrow & \mathrm{ABS} \\ \mathrm{S} & \rightarrow & \mathrm{e} \\ \mathrm{AB} & \rightarrow & \mathrm{BA} \\ \mathrm{BA} & \rightarrow & \mathrm{AB} \\ \mathrm{A} & \rightarrow & \mathrm{a} \\ \mathrm{B} & \rightarrow & \mathrm{b} \end{array} \right\}$$

(4) *Erzeugung von 'abba'*:
a. S $\Longrightarrow$ ABS
b. ABS $\Longrightarrow$ ABABS

c. ABABS $\Longrightarrow$ ABAB
d. ABAB $\Longrightarrow$ ABBA
e. ABBA $\Longrightarrow$ ABbA
f. ABbA $\Longrightarrow$ aBbA
g. aBbA $\Longrightarrow$ abbA
h. abbA $\Longrightarrow$ abba

Bemerkung:
Diese Grammatik erzeugt die Sprache L_0.

(5) *L_0: {x ∈ {a,b}* | x 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).

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

(6) *Beschränkungen für Regeln*:
a. *Typ-0-Grammatiken*:
–
b. *Typ-1-Grammatiken*:
Jede Regel hat die Form αA$\beta \rightarrow \alpha\psi\beta$, wobei $\psi \neq$ e.
c. *Typ-2-Grammatiken*:
Jede Regel hat die Form A $\rightarrow \psi$.
d. *Typ-3-Grammatiken*:
Jede Regel hat die Form A → xB (bzw. A → Bx) oder A → 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.

Grammatiktypen:
- *Typ-0-Grammatiken* ↔ *unbeschränkte Ersetzungssysteme*
- *Typ-1-Grammatiken* ↔ *kontextsensitive Grammatiken*
- *Typ-2-Grammatiken* ↔ *kontextfreie Grammatiken*
- *Typ-3-Grammatiken* ↔ *reguläre Grammatiken* (finite state grammars)

(7) *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 (s.o.); aber auch durch eine (kontextfreie) Typ-2-Grammatik.

Zwei Grammatiken, eine Sprache

(8) *L_0: $\{x \in \{a,b\}^* \mid$ x 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 $=$

$$\left\{ \begin{array}{lcl} S & \rightarrow & ABS \\ S & \rightarrow & e \\ AB & \rightarrow & BA \\ BA & \rightarrow & AB \\ A & \rightarrow & a \\ B & \rightarrow & b \end{array} \right\}$$

(9) *Typ-2-Grammatik*:

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

$$\left\{ \begin{array}{lcl} S & \rightarrow & e \\ S & \rightarrow & aB \\ S & \rightarrow & bA \\ B & \rightarrow & b \\ B & \rightarrow & bS \\ A & \rightarrow & a \\ A & \rightarrow & aS \\ A & \rightarrow & bAA \\ B & \rightarrow & aBB \end{array} \right\}$$

Mögliche Hypothese:
Der am wenigsten mächtige Grammatiktyp, der mit der sprachlichen Evidenz zurecht kommt und es schafft, mit endlichen Mitteln eine unendliche Zahl von Sätzen zu erzeugen, ist der richtige. Also vielleicht reguläre Grammatiken?

(10) *Typ-3-Grammatiken*:
Jede Regel hat die Form A → xB (bzw. A → Bx) oder A → x.
(xB: rechts-regulär; Bx: links-regulär)

(11) a. Karl läuft.
b. Karl läuft und läuft.
c. Karl läuft und läuft und läuft.
d. ...

(12) *Typ-3-Grammatik*:

a. $V_T = \{die,Frau,läuft,und\}$
b. $V_N = \{S,A,B\}$
c. $S \in V_N$
d. $R =$

$$\left\{ \begin{array}{lcl} S & \rightarrow & die\ A \\ A & \rightarrow & Frau\ B \\ B & \rightarrow & läuft\ C \\ B & \rightarrow & läuft \\ C & \rightarrow & und\ B \end{array} \right\}$$

2. Reguläre Grammatiken erfassen natürliche Sprachen nicht

Aber: Es lässt sich zeigen, dass reguläre Grammatiken zur Erfassung natürlicher Sprachen nicht geeignet sind, weil sie keine *Abhängigkeiten* zwischen syntaktischen Bereichen erfassen können.

(13) *Pumping-Lemma* für reguläre Sprachen:
Wenn L eine infinite reguläre Sprache über dem Alphabet Σ ist, dann gibt es Ketten x, y, z ∈ Σ^*, so dass y ≠ e und $xy^nz \in L$, für alle n≥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)

(48) *Pumping-Lemma* für reguläre Sprachen:
Wenn L eine infinite reguläre Sprache über dem Alphabet Σ ist, dann gibt es Ketten x, y, z ∈ Σ^*, so dass y ≠ e und $xy^nz \in L$, für alle n≥0.

Frage:
Ist L_1 in (14) 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 xy^nz in L_1 ist.

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

(15) *Drei mögliche Belegungen für y*:
a. y = eine Anzahl von *a*'s, der eine Anzahl von *b*'s folgt.
b. y = eine Anzahl von *a*'s.
c. y = eine Anzahl von *b*'s.

(16) *Pumping-Lemma* für reguläre Sprachen:
Wenn L eine infinite reguläre Sprache über dem Alphabet Σ ist, dann gibt es Ketten x, y, z ∈ Σ^*, so dass y ≠ e und $xy^nz \in L$, für alle n≥0.

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

1. Fall:
(i) xyz → x = e, y = ab, z = e ⇝ ab
(ii) xyyz → x = e, y = ab, z = e ⇝ *abab

2. Fall:
(i) xyz → x = e, y = a, z = b ⇝ ab

(ii) xyyz → x = e, y = aa, z = b ⇝ *aab

3. Fall:
(i) xyz → x = a, y = b, z = e ⇝ ab
(ii) xyyz → x = a, y = bb, z = e ⇝ *abb
Resultat: L_1 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.
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.

(18) *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.

(19) *Form dieser Sätze*:
$(the + N)^n\ (V_{trans})^{n-1}\ V_{intrans}$

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

Bemerkung:
L_2 ergibt sich aus dem Schnitt von Englisch und der regulären Sprache L_3 = A*B*{died}

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

Frage:
Was sagt das Pumping-Lemma zu L_2?

(22) *Pumping-Lemma* für reguläre Sprachen:
Wenn L eine infinite reguläre Sprache über dem Alphabet Σ ist, dann gibt es Ketten x, y, z ∈ Σ*, so dass y ≠ e und $xy^nz \in L$, für alle n≥0.

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

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.

(24) *Rekursive Komposition*:
a. missiles
b. anti-missile missiles
c. [anti-[anti-missile] missile] missiles
d. [anti-[anti-[anti-missile] missile] missile] missiles
e. *anti-missiles
f. *anti-anti-missiles

(25) Struktur:
$anti^n\ missile^{n+1}$
(n *anti*'s followed by n+1 *missile*'s)

Beobachtung:
Wieder ist das Problem, dass reguläre Grammatiken keine Möglichkeit haben, sich die Zahl des einen Bereichs zu merken und mit der Zahl des anderen Bereichs zu vergleichen.
Noch ein Beispiel: Artikelreplikation im Deutschen.

(26) a. Ich danke dem Kind
b. Ich danke dem dem Mann das Buch gebenden Kind
c. Ich danke dem dem dem Kanzler dankenden Mann das Buch gebenden Kind
d. Ich danke dem dem dem dem Präsidenten vertrauenden Kanzler dankenden Mann das Buch gebenden Kind

3. Kontextfreie Grammatiken erfassen natürliche Sprachen nicht 1: Shieber (1985)

Natürliche Sprachen sind de facto nicht kontextfrei: *Überkreuzende Dependenzen* im Schweizerdeutschen (Shieber (1985); Beweis über Pumping-Lemma für kontextfreie Sprachen).

(27) Jan säit, das mer d'chind$_1$ em Hans$_2$ es huus$_3$ lönd$_1$ hälfed$_2$ aastricht$_3$ lassen
Jan sagt dass wir [die Kinder [dem Hans [das Haus anstreichen] helfen]]

(28) Nicht-kontextfreie Sprachen:
a. $L_1 = \{a^n b^n c^n \mid n \geq 0\}$
b. $L_2 = \{a^n b^m c^n d^m \mid n \geq 0\}$ (schweizerdeutsches Muster)

(29) *Pumping-Lemma* für kontextfreie Sprachen:
Wenn L eine infinite kontextfreie Sprache ist, dann gibt es eine Konstante K, so dass jede Kette w in L, die länger ist als K, in Teilketten w = uvxyz zerlegt werden kann, so dass v und y nicht beide leer sind und $uv^ixy^iz \in L$, für alle $i \geq 0$.

Die Zahl der Akkusativ-NPs entspricht der Zahl der Akkusativ zuweisenden Verben (ebenso für den Dativ); und (abgesehen von "es huus") gehen alle Akkusativ-DPs allen Dativ-DPs voran, und alle Akkusativ zuweisenden Verben allen Dativ zuweisenden Verben.

4. Kontextfreie Grammatiken erfassen natürliche Sprachen nicht 2: Culy (1985)

Hintergrund (Culy (1985)): Bambara hat zwei morphologische Operationen, die miteinander kombiniert werden können und so zu Mustern wie $a^mb^na^mb^n$ führen. Damit ist die Sprache, die Ketten von Morphemen im Bambara involviert, nicht kontextfrei (und daher reichen nur kontextfreie Regeln in der morphologischen Komponente der Grammatik des Bambara nicht aus, um alle Daten korrekt abzuleiten).

(30) a. **N-o-N-Bildung**:
wulu o wulu
Hund o Hund 'welcher Hund auch immer'

b. **N+V_t+la-Bildung** (Komposition plus Derivation):
wulu nyini la
Hund such er 'Hundesucher'

c. **Kombination**:
(i) wulu nyini la^m filè la^n o wulu nyini la^m filè la^n
Hund such er beobacht er o Hund such er beobacht er
(ii) *wulu nyini la o wulu filè la
Hund such er o Hund beobacht er

Beobachtung:

- Das sieht aus wie sequentielle Nominalreduplikation im Deutschen: *Kind auf Kind*, *Reisesser auf Reisesser*, *Hundesucherbeobachter auf Hundesucherbeobachter*.
- Möglicherweise sollten die beiden Ausdruckstypen einheitlich analysiert werden, per Reduplikation (mit Konsequenzen für das Argument gegen Kontextfreiheit).

Chomsky, Noam (1957): *Syntactic Structures*. Mouton, The Hague and Paris.
Culy, Christopher (1985): The Complexity of the Vocabulary in Bambara, *Linguistics and Philosophy* 8, 345–351.
Shieber, Stuart (1985): Evidence Against the Contextfreeness of Natural Language, *Linguistics and Philosophy* 8, 333–343.