Grammatikformalismen und ihre Parskomplexität

1. Sitzung: Organisatorisches und kontextfreie Grammatiken

Daniel Quernheim

Institut für Linguistik, Universität Potsdam
Grammatikformalismen und ihre Parskomplexität,
Thomas Hanneforth

22. April 2009

Thema des Seminars

- Grammatikformalismen
  - linguistische Motivation
  - formale Eigenschaften:
    Abgeschlossenheitseigenschaften, Normalformen etc.
  - Parsingalgorithmen
- Wir betrachten nur Grammatikformalismen, die in polynomieller Zeit parsbar sind!
- Das schließt insbesondere LFG, HPSG etc. aus

Themen im Einzelnen

- Kontextfreie Sprachen
- Mild kontextsensitive Sprachen
- Combinatory categorial grammar (CCG)
- Range concatenation grammar (RCG)
- Synchrone CFGs für Automatische Übersetzung
- Minimalismus
- Baumautomaten
- Vorschläge für weitere Themen sind willkommen!

Linguistische Motivation

Beobachtung:
Menschliche Sprache ist rekursiv aus Konstituenten aufgebaut.

Beispiel:
ein Käfer auf den Blättern der Pflanze, die auf der Fensterbank steht

Einige Eigenschaften kontextfreier Phrasenstruktur

- generiert kontinuierliche Phrasen
- kann projektive Dependenzstrukturen wiedergeben

Kontextfreie Grammatiken

Kontextfreie Sprachen werden erzeugt von Kontextfreien Grammatiken:

$$G = \langle N, \Sigma, P, S \rangle$$

- N eine endliche Menge Nichtterminale
- Σ ein endliches Alphabet von Terminalen
- $S \in N$ das Startsymbol
- P eine endliche Menge von Produktionsregeln der Form

$$A \rightarrow \alpha \quad \text{mit } A \in N,\ \alpha \in (N \cup \Sigma)^*$$

Der Vollständigkeit halber: kontextfreie Sprachen werden von Kellerautomaten erkannt.

Pumping Lemma

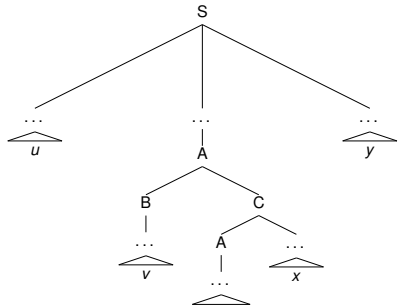
Wenn L kontextfrei ist, gibt es $n \in \mathbb{N}$, so dass gilt:

Für jedes $z \in L$ mit $|z| \geq n$ gibt es $u, v, w, x, y \in \Sigma^*$ mit:

1. $z = uvwxy$
2. $|vx| \geq 1$
3. $|vwx| \leq n$
4. $\forall i \geq 0 : uv^iwx^i \in L$

Pumping Lemma (Beweisidee)

Wenn $|z| \geq 2^{|N|}$, dann muss im Ableitungsbaum ein Nichtterminal mehrfach vorkommen (Schubfachprinzip):



Anwendung des Pumping Lemmas

Mit dem Pumping Lemma kann man zeigen, dass bestimmte Sprachen nicht kontextfrei sind:

- $a^nb^nc^n$ ist nicht kontextfrei
- $ww, w \in \Sigma^*$ ist nicht kontextfrei
- $a^mb^nc^md^n$ ist nicht kontextfrei

Ein Pumping Lemma existiert auch für andere Sprachklassen (bekannt z.B. für reguläre Sprachen)

Abgeschlossenheitseigenschaften

Kontextfreie Sprachen sind abgeschlossen unter

- Vereinigung
- Verkettung
- Sternhülle
- Substitution
- Homomorphismen

Kontextfreie Sprachen sind nicht abgeschlossen unter

- Schnitt
- Komplementbildung

Vereinigung

Seien L_1 und L_2 zwei kontextfreie Sprachen mit den Grammatiken $G_1 = \langle N_1, \Sigma_1, P_1, S_1 \rangle$ und $G_2 = \langle N_2, \Sigma_2, P_2, S_2 \rangle$, wobei die Alphabete disjunkt sind.

Dann erzeugt:

- $G = \langle N_1 \cup N_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 \mid S_2\}, S \rangle$ die Sprache $L_1 \cup L_2$.

Beweis.

- $w \in L_1$ impliziert $S \Rightarrow S_1 \Rightarrow^* w$.
- dito für L_2
- $w \in L(G)$. Dann beginnt eine Derivation von w mit $S \Rightarrow S_1$ oder $S \Rightarrow S_2$. Also folgt: $w \in L_1 \vee w \in L_2$, also $w \in L_1 \cup L_2$.

□

Verkettung

Seien L_1 und L_2 zwei kontextfreie Sprachen mit den Grammatiken $G_1 = \langle N_1, \Sigma_1, P_1, S_1 \rangle$ und $G_2 = \langle N_2, \Sigma_2, P_2, S_2 \rangle$, wobei die Alphabete disjunkt sind.

Dann erzeugt:

- $G = \langle N_1 \cup N_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S \rangle$ die Sprache $L_1 \cdot L_2$.

Beweis.
Sehr einfach. □

Sternhülle

Sei L eine kontextfreie Sprache mit der Grammatik $G = \langle N, \Sigma, P, S \rangle$.

Dann erzeugt:

- $G' = \langle N \cup \{S'\}, \Sigma, P \cup \{S' \to S'S \mid \epsilon\}, S' \rangle$ die Sprache L^*.

Beweis.
Sehr einfach. □

Substitution und Homomorphismen

Definition (Substitution)

Eine Substitution ist eine Abbildung $f : \Sigma \to \Delta^*$, also von Symbolen auf Wörter.

1. Verallgemeinerung (von Wörtern auf Wörter):

- $f(\epsilon) = \epsilon$
- $f(x\alpha) = f(x)f(\alpha)$

2. Verallgemeinerung (von Sprachen auf Sprachen):

$$f(L) = \bigcup_{w \in L} f(w)$$

Definition (Homomorphismus)

Ein Homomorphismus ist eine Substitution h, so dass $h(a)$ immer nur ein Wort enthält.

Substitution und Homomorphismen

Beispiel für Substitution

- $f(0) = a$
- $f(1) = b^*$
- Dann gilt: $f(010) = ab^*a$.
- Wenn $L = 0^*(0+1)1^*$, dann ist $f(L) = a^*(a+b^*)(b^*)^* = a^*b^*$.

Beispiel für Homomorphismus

- $h(0) = aa$
- $h(1) = aba$
- Dann gilt: $h(010) = aaabaaa$.
- Wenn $L = (01)^*$, dann ist $h(L) = (aaaba)^*$.

Substitution und Homomorphismen

- Sei L eine kontextfreie Sprache mit der Grammatik $G = \langle N, \Sigma, P, S \rangle$.
- Für jedes $a \in \Sigma$ sei L_a eine kontextfreie Sprache mit der Grammatik $G_a = \langle N_a, \Sigma_a, P_a, S_a \rangle$, gegeben durch $f(a) = L_a$.

Dann erzeugt:

$$G' = \langle N \cup \bigcup_{a \in \Sigma} N_a, \bigcup_{a \in \Sigma} \Sigma_a, P', S \rangle$$

mit

$$P' = \bigcup_{a \in \Sigma} P_a \cup \{A \to \alpha : A \to \beta \in P \text{ und jedes } a \text{ wurde durch } S_a \text{ ersetzt.}\}$$

die Sprache $f(L)$.

Schnitt und Komplementbildung

Gegenbeispiel

- $L_1 = \{a^i b^j c^j : i \geq 1 \wedge j \geq 1\}$ und
- $L_2 = \{a^j b^j c^j : i \geq 1 \wedge j \geq 1\}$
- sind beide kontextfrei, aber der Schnitt $L_1 \cap L_2 = \{a^i b^i c^i : i \geq 1\}$ ist nicht kontextfrei.

Beweis.

- $L_1 = a^i b^i \cdot c^j$. Da $a^i b^i$ kontextfrei ist und c^j regulär, ist L_1 kontextfrei. Analog für L_2.
- Pumping Lemma: $L_1 \cap L_2$ nicht kontextfrei.

□

Daraus folgt auch: Kontextfreie Sprachen unter Komplementbildung nicht abgeschlossen, denn $L_1 \cap L_2 = \overline{L_1 \cup L_2}$ (Widerspruch).

Entscheidungsprobleme

Entscheidbar für kontextfreie Sprachen:

- Wortproblem: $w (\in \Sigma^*) \in L\,?$
- Leerheitsproblem: $L = \emptyset\,?$
- Endlichkeitsproblem: $|L| < \infty\,?$

Entscheidungsprobleme

Nicht entscheidbar für kontextfreie Sprachen:

- Äquivalenz: $L_1 = L_2\,?$
- Allsprache: $L_1 = \Sigma^*\,?$
- Teilmenge: $L_1 \subseteq L_2\,?$
- Schnitt: $L_1 \cap L_2 = \emptyset\,?$

Normalformen

Wozu Normalformen?

- Normalformen können Beweise vereinfachen.
- Chomsky-Normalform:
  jede Regel hat die Form $A \to BC$ oder $A \to b$
- erweiterte CNF:
  $S \to \epsilon$ zugelassen und S wird von keiner anderen Regel abgeleitet
- Greibach-Normalform:
  jede Regel hat die Form $A \to bC_1 \dots C_n$

Parsing kontextfreier Sprachen

Bekannte Algorithmen

- CYK-Algorithmus (Cocke, Kasami, Younger)
  (nur für Grammatiken in Chomsky-Normalform)
- LR(k)-Algorithmus (Knuth, Tomita)
  (nur für LR(k)-Grammatiken)
- GLR-Parsing (Tomita)
- Chart-Parsing (Earley)
- und andere...

CYK-Algorithmus

- Voraussetzung: Grammatik in Chomsky-Normalform
- löst das Wortproblem in $O(n^3)$
- findet außerdem für jedes Teilwort die Nichtterminale

Teil 1:

for $i = 1$ to n **do**
  for all $(A \to a) \in P$ **do**
    if $a = w_i$ **then**
      $N_{i,i} \leftarrow N_{i,i} \cup \{A\}$
    end if
  end for
end for

CYK-Algorithmus (2)

Teil 2:

for $i = 2$ to n **do**
  for $j = 1$ to $n - i + 1$ **do**
    for $k = 1$ to $i - 1$ **do**
      for all $(A \to BC) \in P$ **do**
        if $B \in N_{j,k} \wedge C \in N_{j+k,i-k}$ **then**
          $N_{j,i} \leftarrow N_{j,i} \cup \{A\}$
        end if
      end for
    end for
  end for
end for
if $S \in N_{1,n}$ **then**
  return true
end if

Korrektheit von CYK

Die Korrektheit des Algorithmus lässt sich mit vollständiger Induktion über die Länge l der Teilwörter beweisen. Voraussetzung: $|w| \geq 1$ (CNF).

Beweis.

- Induktionsanfang: $l = 1$. Für alle Nichtterminale findet Teil 1 des Algorithmus alle Ableitungen $A \Rightarrow w_{i,i+l}$ g.d.w. $A \to w_{i,i+l} \in P$.
- Induktionssschluss von l auf $l + 1$. $A \Rightarrow w_{i,i+l}$ g.d.w. es ein k mit $1 \leq k < l$ gibt, so dass $B \Rightarrow w_{i,i+k}$ und $C \Rightarrow w_{i+k+1,i+l}$. Da $k < l$ und $l - k < l$, können wir die Induktionsvoraussetzung anwenden.

□

Earley-Parsing

- einer von vielen Algorithmen für beliebige kontextfreie Grammatiken
- repräsentiert den Parsefortschritt auf einer Chart in *dotted rules*:

$$\text{NP} \to \text{Det} \bullet \text{N}\ [1, 2]$$

  (an Position 1 könnte eine NP beginnen, von der bereits Det erkannt wurde und ab Position 2 ein N vermutet wird)
- hat eine Laufzeit von $O(n^3)$, für nicht-ambige Grammatiken $O(n^2)$

Earley-Parsing

Der Algorithmus besteht aus drei Komponenten:

- Predictor: erzeugt neue *dotted rules* durch Top-Down-Expansion.
  Beispiel: $\text{S} \to \bullet\ \text{VP}\ [0, 0]$ wird expandiert zu $\text{VP} \to \bullet\ \text{V}\ [0, 0]$ und $\text{VP} \to \bullet\ \text{V}\ \text{NP}\ [0, 0]$.
- Scanner: liest Terminale von der Eingabe und wendet die passenden *dotted rules* an
  Beispiel: Aus $\text{VP} \to \bullet\ \text{V}\ \text{NP}\ [0, 0]$ wird $\text{VP} \to \text{V} \bullet \text{NP}\ [0, 1]$, wenn das Zeichen an Position 0 von der Kategorie V ist.
- Completer: verknüpft Spannen, die bereits erkannt wurden, zu größeren Spannen
  Beispiel: $\text{VP} \to \text{V} \bullet \text{NP}\ [0, 1]$ und $\text{NP} \to \text{Det}\ \text{N} \bullet [1, 3]$ können kombiniert werden zu $\text{VP} \to \text{V}\ \text{NP} \bullet [0, 3]$.

Nicht-Kontextfreiheit in natürlichen Sprachen

Schweizerdeutsch (Shieber 1985)

...Jan säit das mer d'chind em Hans es huus lönd hälfe aastriche.

- Kasuskongruenz: Dativ und Akkusativ
- Abfolge: Dativ-NPs vor Akkusativ-NPs
- Kreuzeinbettung (cross-serial dependencies)
- lässt sich mit Substitution auf $a^m b^n c^m d^n$ abbilden, was nicht kontextfrei ist.

Nicht-Kontextfreiheit in natürlichen Sprachen

Deutsch (Groenink 1997)

- ..., dass Fred Julia Frank schwimmen helfen ließ und ertrinken lassen sah.
- ..., dass Hans Fred Julia Frank schwimmen helfen hören ließ und ertrinken lassen sehen ließ.