Chomsky-Hierarchie

Sprachen, Grammatiken, Automaten

Eduard Mehofer

Fakultät für Informatik

Währinger Straße 29

Universität Wien

Definition einer Sprache (1)

Sei Σ ein Alphabet. Eine formale Sprache L ist als eine Teilmenge $L \subseteq \Sigma^*$ definiert. Wie lassen sich nun formale Sprachen präzise beschreiben? - Folgende Methoden sind existieren:

1. Auflistung/Aufzählung aller Sprachelemente

Die explizite Auflistung aller Sprachelemente ist nur für endliche Sprachen (mit relativ wenigen Elementen) möglich und daher für die meisten in der Praxis interessanten Sprachen irrelevant.

2. Spezifikation eines formalen Ausdrucks

Ein Beispiel für diese Methode sind reguläre Ausdrücke, die beliebige reguläre Sprachen beschreiben können. Zum Beispiel kann man die Menge der Bezeichner einer Programmiersprache durch $L = B(B+Z)^*$ darstellen, wenn B die Menge der Buchstaben und Z die Menge der Ziffern repräsentiert.

Definition einer Sprache (2)

3. Grammatiken

Grammatiken lassen sich zur (rekursiven) Erzeugung der Sätze einer Sprache benutzen. Es wird ein Verfahren angegeben, das systematisch alle Sätze einer Sprache generiert.

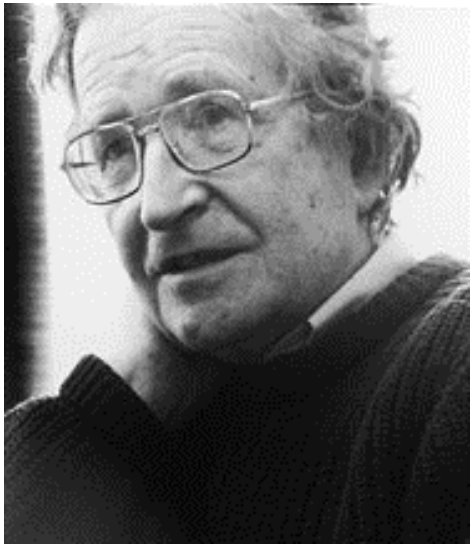
4. Automaten

Automaten lassen sich zur Erkennung der Sätze einer Sprache verwenden. Es wird ein Verfahren spezifiziert, das für jedes Wort w entscheidet, ob w in der Sprache enthalten ist oder nicht. Die von einem Automaten A akzeptierte Sprache L(A) ist die Menge aller akzeptierten Wörter.

Chomsky-Hierarchie

Die Chomsky Hierarchie (nach Noam Chomsky; 1956,1959) legt eine Hierarchie von vier Sprachklassen fest und formalisiert die Beziehung zu Grammatik- und Automatenklassen.

Noam Chomsky, geb. 1928,
bekannter amerikanischer Linguist,
Prof. emer. MIT.
Populär: Medienkritik, polit. Engagement



Grammatiken der Chomsky Hierarchie

Eine Chomsky Grammatik hat die Form $G = (N, \Sigma, P, S)$, wo N und Σ die Alphabete der Nichtterminal- bzw Terminalsymbole darstellen, P die Regelmenge ist und $S \in N$ das Startsymbol darstellt. Die einzelnen Grammatikklassen werden durch die Form ihrer Regeln unterschieden:

- Typ-0 Grammatik (allg. Gr.): siehe folg. Folien.
- Typ-1 Grammatik (kontextsensitiv): siehe folg. Folien.
- Typ-2 Grammatik (kontextfrei): bekannt
  - $A \rightarrow \alpha$ mit $A \in N$, $\alpha \in \Gamma^*$ ($\Gamma = N \cup \Sigma$)
- Typ-3 Grammatik (regulär): bekannt
  - rechtslinear: $A \rightarrow x$ oder $A \rightarrow xB$, mit $A, B \in N$ und $x \in \Sigma^*$.
  - linkslinear: $A \rightarrow x$ oder $A \rightarrow Bx$, mit $A, B \in N$ und $x \in \Sigma^*$.

Automaten der Chomsky Hierarchie

Den vier Grammatik- und Sprachklassen der Chomsky Hierarchie entsprechen vier Klassen von Automaten. Die 4 Automatenklassen heißen:

- Typ-0: Turingmaschinen (siehe folg. Folien)
- Typ-1: Linear beschränkte Automaten (siehe folg. Folien)
- Typ-2: Kellerautomaten (siehe folg. Folien)
- Typ-3: Endliche Automaten (bekannt)

Sätze zur Chomsky Hierarchie

- **Typ-i Automaten ($0 \leq i \leq 3$) erkennen genau die von Typ-i Grammatiken generierten Sprachen.**

- **Eine Sprache L die von einer Typ-i Grammatik G erzeugt wird, i.e. L=L(G), bzw. von einem Typ-i Automaten M akzeptiert wird, i.e. L=L(M), heisst Typ-i Sprache.**

Hierarchiesatz:

Bezeichne L_i, $0 \leq i \leq 3$, die Familie der Chomsky Typ-i Sprachen. Dann gilt (echte Teilmengenbeziehungen; L_3 echte Teilmenge von L_2 usw., d.h. L_2 umfasst mehr Sprachen als L_1):

$$L_3 \subset L_2 \subset L_1 \subset L_0$$

Sätze

1. Zu jedem nichtdeterministischen endlichen Automaten A, gibt es einen deterministischen Automaten A´ mit L(A) = L(A´).

2. Zu jedem regulären Ausdruck α existiert ein endlicher Automat A mit L(A) = L(α).

3. Zu jedem endlichen Automaten A existiert ein regulären Ausdruck α mit L(α) = L(A).

4. Ist G eine reguläre Grammatik, dann gibt es einen endlichen Automaten A mit L(A) = L(G).

5. Ist A ein endlicher Automat, dann existiert eine reguläre Grammatik G mit L(G) = L(A).

Bemerkungen zur Hierarchieeigenschaft

- **Die Sprache**

$$\{ a^n b^n \mid n \geq 1 \}$$

ist kontextfrei, aber nicht regulär.

(Kontextfreie Grammatik: $G = (\{S\}, \{a, b\}, \{S \rightarrow aSb \mid ab \} , S)$.)

- **Die Sprache**

$$\{ a^n b^n c^n \mid n \geq 1 \}$$

ist kontextsensitiv, aber nicht kontextfrei.

Wann ist eine Sprache nicht regulär bzw. kontextfrei? (Beweise mit Pumping Lemma)

Theorem: Pumping Lemma für reguläre Sprachen

Sei L eine reguläre Sprache. Dann gibt es ein **n** , so dass jedes Wort $w \in L$ mit $|w| \geq \mathbf{n}$ in $w = x\,y\,z$ zerlegt werden kann, wobei

1. $y \neq \varepsilon$
2. $|xy| \leq \mathbf{n}$
3. $x\,y^{\mathbf{i}}\,z \in L$ für alle $i \geq 0$

Theorem: Pumping Lemma für kontextfreie Sprachen

Sei L eine kontextfreie Sprache. Dann gibt es ein **n** , so dass jedes Wort $w \in L$ mit $|w| \geq \mathbf{n}$ in $w = u\,v\,w\,x\,y$ zerlegt werden kann, wobei

1. $vx \neq \varepsilon$
2. $|vwx| \leq \mathbf{n}$
3. $u\,v^{\mathbf{i}}\,w\,x^{\mathbf{i}}\,y \in L$ für alle $i \geq 0$

Typ-0 Grammatiken

Definition: Eine **Typ-0 Grammatik** ist ein Quadrupel $G = (N, \Sigma, P, S)$

wobei N, Σ *und* S die gleiche Bedeutung wie bei KFG haben
und wir das Gesamtalphabet ebenfalls mit Γ bezeichnen.

Für die Regeln gilt jedoch:

$$P \subseteq \Gamma^* N \Gamma^* \times \Gamma^*$$

Regeln einer Typ-0 Grammatik sind also von der Form $\alpha \rightarrow \beta$, wobei α und β Worte über dem Gesamtalphabet sind, mit der Bedingung, dass α mindestens ein Nichtterminalsymbol enthält. Wie bei KFG lassen sich Regeln als Vorschriften für Textersetzungen in Worten aus Γ^* interpretieren, wobei allerdings Teilworte (die als linke Seite einer Regel auftreten) und nicht nur (wie bei KFG) einzelne Nichtterminalsymbole ersetzt werden können.

Beispiel:

Gegeben Sprache $L = \{ a^{2^i} \mid i \geq 1 \}$.

Typ-0 Grammatik G mit L(G) = L ist gegeben durch:
G = ({ S, A, B, C, D, E }, { a }, P, S)

P = {
1: $S \rightarrow ACaB$,
2: $Ca \rightarrow aaC$,
3: $CB \rightarrow DB$,
4: $CB \rightarrow E$,
5: $aD \rightarrow Da$,
6: $AD \rightarrow AC$,
7: $aE \rightarrow Ea$,
8: $AE \rightarrow \varepsilon$ }

A und B dienen als linke und rechte Markierungen für das Ende der Satzform. Var. C läuft über die aus a´s bestehende Zeichenkette von A nach B und verdoppelt dabei mit Regel 2) die Anzahl der a´s. Var. D läuft von B zurück nach A. Wenn man terminieren möchte, löscht man mit Regel 4) Var. B, Var. E läuft zurück nach A und löscht A.

$S \Rightarrow ACaB \Rightarrow AaaCB \Rightarrow AaaE \Rightarrow AaEa \Rightarrow AEaa \Rightarrow aa$

oder

$S \Rightarrow ACaB \Rightarrow AaaCB \Rightarrow AaaDB \Rightarrow AaDaB \Rightarrow ADaaB \Rightarrow ACaaB \Rightarrow \ldots$

Kontextsensitive (Typ-1) Grammatiken

Definition: Eine Typ-0 Grammatik heißt **kontextsensitiv (KSG)** oder **Typ-1 Grammatik**, falls jede Regel in P eine der folgenden beiden Formen besitzt:

Entweder

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \alpha \alpha_2$$

wobei $A \in N$, $\alpha_1, \alpha_2 \in \Gamma^*$ und $\alpha \in \Gamma^+$

oder

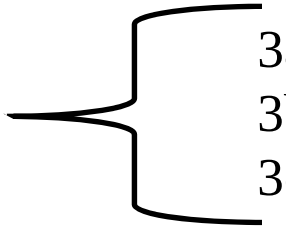
$S \rightarrow \varepsilon$ und S tritt nicht auf der rechten Seite von Regeln auf.

Beispiel – Kontextsensitive Sprache

Beispiel: Kontextsensitive Grammatik für die Sprache $L = \{a^n b^n c^n \mid n>0\}$.

Lösung:
Idee: man erzeugt $\mathbf{a}^n(BC)^n$, alle C´s müssen nach hinten verschoben werden, B wird terminal auf **b** und C wird terminal auf **c** abgeleitet (Kontext in Prod. unterstrichen).

P = {
1: $S \rightarrow \mathbf{a}SBC$,
2: $S \rightarrow \mathbf{a}BC$,
3: $CB \rightarrow BC$,
4: $\mathbf{a}B \rightarrow \mathbf{ab}$,
5: $\mathbf{b}B \rightarrow \mathbf{bb}$,
6: $\mathbf{b}C \rightarrow \mathbf{bc}$,
7: $\mathbf{c}C \rightarrow \mathbf{cc}$
}



3a: $C\underline{B} \rightarrow C_1\underline{B}$
3b: $\underline{C_1}B \rightarrow \underline{C_1}C$
3c: $C_1\underline{C} \rightarrow B\underline{C}$

Produktion 3 muß durch 3a/3b/3c ersetzt werden!

$G = (\{S, B, C, C_1\}, \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}, P, S)$

Problem: Produktion 3 erfüllt nicht unsere Einschränkung für kontextsensitive Sprachen!