Formale Sprachen

(parallele und regulierte Ersetzung)

SoSe 2008 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Formale Sprachen Gesamtübersicht

- Organisatorisches / Einführung
  Motivation / Erinnerung / Fragestellungen

- Diskussion verschiedener Sprachklassen:
  gesteuerte Ersetzungsverfahren
  parallele Ersetzungsverfahren

- Algebraische Ansätze:
  abstrakte Sprachfamilien
  formale Potenzreihen

Parallele Grammatiken

- Lindenmayer-Systeme

- eingeschränkter Parallelismus

Erinnerung: **D0L Systeme**

Einfachster Typus von Lindenmayer-Systemen.

Spezifiziert durch $G = (\Sigma, h, \omega)$, wobei:
Σ Alphabet, $\omega \in \Sigma^*$ *Axiom* (Startwort),
h Morphismus auf Σ, angegeben als $h : \Sigma \to \Sigma^*$, oder als Regelmenge $P = \{a \to w \mid a \in \Sigma, w \in \Sigma^*\}$, die
vollständig (zu jedem $a \in \Sigma$ gibt es mindestens eine Regel mit solcher linken Seite) sein soll und auch
deterministisch (zu jedem $a \in \Sigma$ gibt es höchstens eine Regel mit solcher linken Seite)

Ableitungsbegriff: $x \Rightarrow y$ gdw. $y = h(x)$ (uneingeschränkt parallele Ableitung)

Wie üblich: $L(G) = \{w \in \Sigma^* \mid \omega \stackrel{*}{\Rightarrow} w\}$.

Natürliche Verallgemeinerung: *0L-Systeme*
Fortlassen der “Determinismus-Bedingung”.
Formal bedeutet dies, dass wir zu jeder linken Seite a eine nicht-leere, endliche Teilmenge $h(a)$ assoziieren.
$h : \Sigma \to 2^{\Sigma^*}$ lässt sich wieder als Morphismus
$h : \Sigma^* \to 2^{\Sigma^*}$ begreifen (Komplexprodukt).

Mit $G = (\Sigma, h, \omega)$ gilt dann:
Ableitungsbegriff: $x \Rightarrow y$ gdw. $y \in h(x)$ (uneingeschränkt parallele Ableitung)

Wie üblich: $L(G) = \{w \in \Sigma^* \mid \omega \stackrel{*}{\Rightarrow} w\}$.

Beispiel: Regelmenge $a \to a, a \to aa$, $\omega = a$, $L = \{a^? \mid \ldots\}$
Gibt es äquivalentes D0L-System ?

Weitere 0L-Sprachen

Beispiel: Ist $L = \{w\}$, so sind L und $L \cup \{\lambda\}$ D0L-Sprachen.
Tatsächlich benötigt man dafür nur entweder Regeln $a \to a$ oder $a \to \lambda$.
Beispiel: $L_k = \{a^i \mid i \in \mathbb{N}, i \leq k\}$ sind 0L-Sprachen, aber nur dann auch D0L-Sprachen, wenn $k \leq 1$.

Beweis: Hier brauchen wir beide Arten von Regeln.

Wäre $L_2 = \{\lambda, a, a^2\}$ D0L-Sprache, so müsste $a \to \lambda$ die Regel sein. Gleich welches Axiom gewählt würde, gilt: Läge a in der Sprache, so a^2 nicht, und läge a^2 in der Sprache, so a nicht.

Nicht-0L-Sprachen

Beispiel: $L = \{a, a^3\}$ ist keine 0L-Sprache.
Beweis: Angenommen, es gäbe ein 0L-System $G = (\Sigma, h, \omega)$ für L.
Wir unterscheiden zwei Fälle:

- $\omega = a$. Dann gilt: $a \Rightarrow aaa$, also $a^3 \Rightarrow a^9 \leadsto \lightning$
- $\omega = a^3$. Dann gilt: $aaa \Rightarrow a$, also $a \Rightarrow \lambda$ und $a \Rightarrow a$. Also ist auch $a^3 \Rightarrow a^2 \leadsto \lightning$.

Ähnlich sieht man:
Beispiel: $L = \{a, a^4\}$ ist keine 0L-Sprache.
Komplizierter ist schon:
Beispiel: $\{\lambda, a^5, a^{10}\}$ ist keine 0L-Sprache.
Beispiel: Betrachte $G = (\{a\}, \{a \to a^3, a \to a^4\}, a)$ und $\phi : a \mapsto a^5$, aufgefasst als Morphi. $L = \phi^{-1}(L(G))$ ist keine 0L-Sprache.

Ein komplizierteres Argument

Lemma: Für $L = L(G)$ mit $G = (\{a, b\}, \{a \to a^2, b \to b^3\}, ab)$ gilt:
$L = \{a^{2^n} b^{3^n} \mid n \in \mathbb{N}\}$. Dann ist L^+ keine 0L-Sprache.

Beweis: Angenommen, $G^+ = (\{a, b\}, h, \omega)$ ist 0L-System für L^+.
Diskutiere $\omega = a^{r_1} b^{s_1} \dots a^{r_t} b^{s_t}$. Enthielte jedes nicht-leere $\alpha \in h(a)$ sowohl a'als auch b's, so wäre

$$\max \{\max\{r_i \mid 1 \leq i \leq t\}, 2 \cdot \max\{|\alpha| \mid \alpha \in h(\{a, b\})\}\}$$

↯ zur Struktur von L^+. Da kein Wort in L^+ mit b beginnt, gilt: $h(a) \cap \{b\}^+ = \emptyset$.

Also gibt es ein $\alpha \in h(a)$ mit $\alpha \in \{a\}^*$. Entsprechend gilt für $\beta \in h(b)$: $\beta \in \{b\}^*$.

(A) $\lambda \notin h(a) \cup h(b)$. Gölte o.E. $\lambda \in h(a)$, so nicht $\lambda \in h(b)$, da $\lambda \notin L^+$, d.h., es gäbe ein $\beta \in h(b)$ mit $\beta \in \{b\}^+$; zusammen mit $a \to \lambda$ ließe sich so ein Wort aus ω ableiten, das nicht zu L^+ gehört.

(B) Wegen $\alpha \in h(a)$ für ein $\alpha \in \{a\}^+$ gilt $h(b) \subset \{b\}^+$, denn $\alpha^{2^n} \beta^{3^n} \in L(G^+)$ für alle $n \in \mathbb{N}$.
Entsprechend folgt: $h(a) \subset \{a\}^+$.

(C) Wegen (B) können aus ω nur Wörter abgeleitet werden mit t (abwechselnden) a- und b-Blöcken, ↯ zur Struktur von L^+.

Satz: Die Familie der 0L-Sprachen ist nicht abgeschlossen gegen (1) Vereinigung, (2) Konkatenation, (3) Kleene-Plus, (4) Durchschnitt mit regulären Sprachen, (5) nicht-löschende Morphismen und auch nicht gegen (6) inverse Morphismen. *Anti-AFL*

Beweis: (1) $\{a\}$ und $\{a^3\}$ sind D0L-Sprachen, ihre Vereinigung ist keine 0L-Sprache.

(2) $\{a\}$ und $\{\lambda, a^2\}$ sind 0L-Sprachen, ihre Konkatenation ist keine 0L-Sprache.

(3) siehe Lemma

(4) $\{a^{2^n} \mid n \in \mathbb{N}\}$ ist D0L-Sprache und $L = \{a, a^4\}$ ist regulär, ihr Schnitt ist gleich L und daher nicht 0L-Sprache.

(5) Betrachte $\phi : a \mapsto a^5$, aufgefasst als Morphi. Beispiel oben: $L = \{\lambda, a, aa\}$ ist 0L-Sprache, aber $\phi(L) = \{\lambda, a^5, a^{10}\}$ ist es nicht.

(6) siehe obiges Beispiel

Weitere Sprachen aus 0L-Systemen: E0L-Systeme

Ein *E0L-System* (E: extension, Erweiterung) ist ein Quadrupel $G = (\Sigma, h, \omega, \Delta)$; hierbei ist $U(G) = (\Sigma, h, \omega)$ das *unterliegende* 0L-System, und $\Delta \subseteq \Sigma$ ist das *Terminal-* oder *Zielalphabet*. Die von G beschriebene Sprache ist $L(G) = L(U(G)) \cap \Delta^*$.

Mathematische Motivation:

Zwischen kontextfreien Grammatiken und 0L-Systemen gibt es zwei wesentliche Unterschiede:

(1) Parallele Ableitung und

(2) (Nicht-)Unterscheidung von Terminal- und Nichtterminalzeichen.

E0L-Systeme isolieren den ersteren Unterschied.

Beispiele für E0L-Systeme

Beispiel: $G = (\{S, a, b\}, \{S \to a, S \to b, a \to a^2, b \to b^2\}, S, \{a, b\})$ beschreibt $L(G) = \{a^{2^n}, b^{2^n} \mid n \in \mathbb{N}\}$. Gibt es hierfür ein 0L-System ?

Beispiel: $G = (\{A, a, b\}, \{A \to A, A \to a, a \to a^2, b \to b\}, AbA, \{a, b\})$, $L(G) = \{a^{2^n} b a^{2^m} \mid n, m \in \mathbb{N}\}$.

Lemma: Jede endliche Sprache ist E0L-Sprache.

Synchronisierte E0L-Systeme

Ein E0L-System $G = (\Sigma, h, \omega, \Delta)$ heißt *synchronisiert* gdw. $\forall a \in \Delta \forall \beta \in \Sigma^* : (a \stackrel{*}{\Rightarrow} \beta \rightsquigarrow \beta \notin \Delta^*)$.

Satz: Es gibt einen Algorithmus, der ein vorgelegtes E0L-System G in ein äquivalentes synchronisiertes System $\bar{G}$ umformt. Genauer hat dieses System $\bar{G} = (\Sigma, h, \omega, \Delta)$ folgende Eigenschaften:
(1) $\omega \in \Sigma \setminus \Delta$ (Startsymbol)
(2) Es gibt ein *Fehlersymbol* $F \in \Sigma \setminus \Delta$, sodass für jedes $a \in \Delta \cup \{F\}$ $a \to F$die einzige Produktion für a ist.
(3) Rechte Regelseiten sind entweder Terminalwörter oder das Fehlersymbol, oder sie enthalten weder Terminalzeichen, noch das Fehler- oder Startsymbol.
(4) Aus jedem Nichtterminalzeichen außer dem Fehlersymbol und möglicherweise dem Startsymbol lässt sich ein Terminalwort ableiten.
Mit G ist auch $\bar{G}$ propagierend.

Synchronisierte E0L-Systeme: Anwendung

Satz: Sind L_1 und L_2 E0L-Sprachen, so auch:

$L_1 \cup L_2$,

$L_1 L_2$,

L_1^+

und $L_1 \cap R$ für eine beliebige reguläre Sprache R.

Beweise an der Tafel.

Hinweis: Normalformen sind wichtig !

EP0L versus E0L

Satz: Zu jedem E0L-System $H = (\Sigma, h, S, \Delta)$ kann ein EP0L-System $G = (V, g, [S, \emptyset], \Delta)$ konstruiert werden mit $L(H) \setminus \{\lambda\} = L(G)$.
$\alpha(x)$: Menge der Zeichen, die in x vorkommen.

Beweis: O.E.: H in Normalform, d.h., $S \in \Sigma \setminus \Delta$, und $L(H)$ ist unendlich.

$V = \Sigma \times 2^{\Sigma} \cup \{F\} \cup \Delta$, die zweite Komponente enthält Symbole, die verschwinden sollen während der Ableitung in H. Diese Zeichenmenge rät und überprüft G.

g enthält die folgenden Regeln (und nur diese):

(1) Gilt $b \in h(a)$ mit $b \in \Sigma$, so sei $[b, Z'] \in g([a, Z])$, falls
$Z' \in \sigma(Z)$, d.h.: $\exists x, x' \in \Sigma^* : \alpha(x) = Z \wedge \alpha(x') = Z' \wedge x \Rightarrow_H x'$.

(2) Gilt $b_1 \dots b_k \in h(a)$, $b_i \in \Sigma$, $k \geq 2$, so sei $[b_{i_1}, Z_{i_1}][b_{i_2}, Z_{i_2}] \dots [b_{i_p}, Z_{i_p}] \in g([a, Z])$,
falls $i_0 = 1 \leq i_1 < i_2 < \dots < i_p \leq i_{p+1} = k$, $Z' \in \sigma(Z)$, sowie
$Z_{i_j} = \alpha(b_{i_{j-1}+1} \dots b_{i_j-1} b_{i_j+1} \dots b_{i_{j+1}-1}) \cup Z'$ für $1 \leq j \leq p$.

(3) $a \in g([a, \emptyset])$ für $a \in \Delta$.

(4) $g(X) = \{F\}$ für $X \in \Delta \cup \{F\}$.

Weitere Sprachen aus 0L-Systemen: C0L-Systeme

Es seien Σ und Σ' Alphabete. Eine Abbildung $g : \Sigma \to \Sigma'$ heißt auch *Kodierung*. Kodierungen kann man auch als Morphismen zwischen frei erzeugten Monoiden auffassen.

Ein *C0L-System* ist ein Quadrupel $G = (\Sigma, h, \omega, \phi)$; hierbei ist $U(G) = (\Sigma, h, \omega)$ das *unterliegende* 0L-System, und $\phi : \Sigma \to \Delta$ ist eine Kodierung.
Die von G beschriebene Sprache ist $L(G) = \phi(L(U(G))) \subseteq \Delta^*$.

Biologische Motivation:
Ermöglicht (insbesondere, wenn $|\Delta| < |\Sigma|$) einem beobachteten Zustand mehrere "wirkliche" Zustände zuzuordnen.

Ein **Beispiel** für ein C0L-System für $L = \{a^n b^n c^n \mid n \geq 1\}$:

$$
\begin{aligned}
A &\rightarrow AA', & A' &\rightarrow A' \\
B &\rightarrow BB', & B' &\rightarrow B' \\
C &\rightarrow CC', & C' &\rightarrow C'
\end{aligned}
$$

seien die Regeln eines 0L-Systems mit Axiom ABC welches, zusammen mit der Kodierung $A, A' \mapsto a$, $B, B' \mapsto b$, and $C, C' \mapsto c$ die Sprache L beschreibt. Genauer gilt für die n-Schritt Ableitung:

$$ABC \Rightarrow^n A(A')^n B(B')^n C(C')^n$$

Kodierungen und Erweiterungen

Lemma: Jede C0L-Sprache ist eine E0L-Sprache.
Dies gilt auch für propagierende Systeme.

Beweis: Es sei $G = (\Sigma, h, \omega, \phi)$ mit $\phi : \Sigma \to \Delta$ ein C0L-System. O.E. sei $\Sigma \cap \Delta = \emptyset$.
Es gibt ein EP0L-System $\bar{G} = (\Theta, \bar{g}, S, \Sigma)$ mit $L(\bar{G}) = L(U(G)) \setminus \{\lambda\}$. O.E. sei $\bar{G}$ synchronisiert und $\Theta \cap \Delta = \emptyset$.
Betrachte $H = (\Theta \cup \Delta, g, \Delta)$, wobei g aus folgenden Regeln besteht (und nur diesen):
$\{X \to \phi(x) \mid x \in \Sigma^*, X \to x \in \bar{g}\} \cup \{X \to F \mid X \in \Delta\} \cup \{X \to x \mid X \to x \in \bar{g}, x \notin \Sigma^*\}$.
Es gilt: $L(H) = \phi(L(\bar{G})) = \phi(L(U(G)) \setminus \{\lambda\}) = \phi(L(U(G))) \setminus \{\lambda\}$.
Falls $\lambda \in L(G)$, muss H noch leicht modifiziert werden.

Beobachte: Welche früheren Aussagen wurden verwendet ?
Aufgabe: “Einfaches” E0L-System für $L = \{a^n b^n c^n \mid n \geq 1\}$.

Kodierungen und Erweiterungen

Ohne Beweise teilen wir noch folgende Ergebnisse mit:

Satz: Eine Sprache ist C0L-Sprache genau dann, wenn sie E0L-Sprache ist.

Das bedeutet: Ein mathematisch/linguistisch gut motivierter Begriff stimmt im Wesentlichen mit einem biologisch gut motivierten überein.

Satz: Die Familie der CP0L-Sprachen ist eine echte Teilfamilie der EP0L-Sprachen.

Trennsprache: $L = \{a^{3^n}, b^n c^n d^n \mid n \geq 1\}$. Intuition ?

Wachstumsmuster und Interaktion

Bei sog. (i, j)*L Systemen* (oder kurz *IL Systemen*, falls die Zahlen i und j bel.), werden i Zeichen “links” und j Zeichen “rechts” von der Ersetzungsstelle mit in Betracht gezogen, was die Anwendbarkeit von Regeln angeht.

Gibt es links von der Ersetzungsstelle weniger als $i > 0$ Zeichen in der augenblicklichen Satzform, so nimmt man an, das Sonderzeichen # werde “gesehen”. Genauso verfährt man am rechten Rand.

Ein Beispiel für DIL Systeme: Gabors Faultier

Axiom ad, Regeln:

	a	b	c	d
#	c	b	a	d
a	a	b	a	d
b	a	b	a	d
c	b	c	a	ad
d	a	b	a	d

Die Ableitung geht wie folgt:

$$ad \Rightarrow cd \Rightarrow aad \Rightarrow cad \Rightarrow abd \Rightarrow cbd \Rightarrow acd \Rightarrow caad \Rightarrow abaad$$

Gabor's sloth zeigt logarithmisches Wachstum (warum?).

Wachstumsfunktionen

Zu einem vorgelegten DIL System $G = (\Sigma, h, \omega)$ gibt seine *DIL Wachstumsfunktion*

$$g_G : n \mapsto |w_n| \quad \text{for} \quad \omega \Rightarrow^n w_n$$

die Länge des nach n Schritten abgeleiteten Wortes an.

Lemma: Kein DIL System wächst schneller als exponentiell.

Welche Arten von Funktionen können aber durch DIL Systeme beschrieben werden ?

1. exponentielles Wachstum: Betrachte D0L System mit Regel $a \to a^2$;
2. polynomielles Wachstum: $a \to ab, b \to b$ erzeugen, beginnend mit a, ab^n;
3. logarithmisches Wachstum: siehe Gabors Faultier.

Wachtumsmuster ohne Interaktion

g_G hängt nicht von der Abfolge der Zeichen im Axiom bzw. in rechten Regelseiten ab.
Wichtig hingegen: Wie viele Zeichen welcher Art kommen vor ?
Diese Information kann man im sog. *Parikh (Zeilen) Vektor* des Axioms und in der *Wachstumsmatrix* der Regeln ablegen.

Quadratisches Wachstum beobachten wir für ein D0L System auf den folgenden Folien:

Wachstumsfunktionen am Beispiel D0L System

$$G = (\{a, b, c\}, \{a \to abc^2, b \to bc^2, c \to c\}, a).$$

Wachstumsmatrix:

$$M_G := \begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

Die erste Zeile von M_G ist der Parikh-Vektor von abc^2, die zweite der von bc^2, due dritte gehört zur Regel $c \to c$.

Wie sieht die Matrix der Parikhvektoren aus von den Wörtern, die von a, b bzw. c aus in zwei Ableitungsschritten erzeugt werden können ?
Diese Information erhält man einfach durch Quadrieren von M_G:

$$M_G^2 = \begin{pmatrix} 1 & 2 & 6 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}$$

Beispielableitung hierzu:

$$a \Rightarrow abcc \Rightarrow abccbcccc \quad \text{und} \quad (1\;0\;0)\, M_G^2 = (1\;2\;6) \quad \checkmark$$

Algebra und Wachstumsfunktionen gehören zusammen:

Ein klassisches Ergebnis aus der Matrizentheorie ist der Satz von Cayley-Hamilton: Jede Matrix befriedigt ihre eigene charakteristische Gleichung (Polynom). Daraus folgt:

$$M_G^n = c_1 M_G^{n-1} + c_2 M_G^{n-2} + \cdots + c_n M_G^0,$$

was die folgende Rekursion für die Wachstumsfunktion g_G liefert:

$$g_G(i+n) = c_1 g_G(i+n-1) + c_2 g_G(i+n-2) + \cdots + c_n g_G(i)$$

($i \geq 0$ bel.). Das gestattet oft eine explizite Bestimmung von g_G.

Algebra und Wachstumsfunktionen: Nochmal unser Beispiel

$$M_G^3 = \begin{pmatrix} 1 & 3 & 12 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{pmatrix} = c_1 \begin{pmatrix} 1 & 2 & 6 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix} + c_2 \begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} + c_3 \mathbb{I},$$

$\mathbb{I}$: Einheitsmatrix. Mit etwas Rechnen folgt:

$$c_1 = 3, \quad c_2 = -3 \text{ und } \quad c_3 = 1.$$

$$g_G(i+3) = 3g_G(i+2) - 3g_G(i+1) + g_G(i)$$

mit Anfangswerten

$$g_G(0) = 1, \quad g_G(1) = 4 \text{ und } \quad g_G(2) = 9.$$

$\rightsquigarrow g_G(i) = i^2$ durch Überprüfen von

$$(i+3)^2 = i^2 + 6i + 9 = 3(i+2)^2 - 3(i+1)^2 + i^2.$$

Algebra und Wachstumsfunktionen: Weitere Folgerungen

Lemma: Ist g eine D0L Wachstumsfunktion mit beliebig langen konstanten Intervallen, d.h. formaler:

$$\forall k \exists i : g(i) = g(i+1) = \cdots = g(i+k),$$

so ist g is *schließlich konstant*.

Gabors Faultier besitzt jedoch beliebig lange konstante Intervalle, d.h.:
Satz: Es gibt DIL Wachstumsfunktionen, die keine D0L Wachstumsfunktionen sind.

Man kann jetzt auch ganz eigentümliche Entscheidbarkeitsfragen stellen wie:
Ist die Wachstumsfunktion eines vorgelegten D0L Systems polynomiell ?