

Formale Sprachen (parallele und regulierte Ersetzung)

SoSe 2007 in Trier

Henning Fernau

Universität Trier

fernau@informatik.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

DOL 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 $h : \Sigma \rightarrow \Sigma^*$, oder als Regelmenge $P = \{a \rightarrow w \mid a \in \Sigma, w \in \Sigma^*\}$, 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: $L(G) = \{w \in \Sigma^* \mid \omega \xRightarrow{*} w\}$.

D0L Systeme: Beispiele

$$h(a) = a^2, G = (\{a\}, h, a), L(G) = \dots$$

$$h(X) = aXb, h(a) = a, h(b) = b, G = (\{a, b, X\}, h, X), L(G) = \dots$$

Gibt es D0L-System für $\{a^n X b^n Y c^n \mid n \geq 0\}$?

Beobachtung $L(G)$ kann nicht leer sein !

Folgen

Jedes D0L-System $G = (\Sigma, h, \omega)$ definiert eine Abbildung $E(G) : \mathbb{N} \rightarrow \Sigma^*$, $n \mapsto h^n(\omega)$, die *D0L-Folge*.

Klar: $L(G) = \{E(G)(n) \mid n \geq 0\}$.

Beispiel: $G = (\{a, b\}, \{a \rightarrow a, b \rightarrow ab\}, ab)$. $E(G)(n) = a^{n+1}b$.

Ein Morphismus $h : \Sigma^* \rightarrow \Delta^*$ heißt *nicht-löschend* gdw. $\forall w \in \Sigma^* : h(w) \neq \lambda$.

Hilfssatz: $h : \Sigma^* \rightarrow \Delta^*$ ist nicht-löschend gdw. $\forall w \in \Sigma : h(w) \neq \lambda$.

D0L-Systeme mit nicht-löschenden Morphismen heißen *propagierend* oder kurz *PD0L-Systeme*.

Wachstumsfunktionen

Die Wachstumsfunktion $f_G : \mathbb{N} \rightarrow \mathbb{N}$ eines D0L-Systems $G = (\Sigma, h, \omega)$ ist gegeben durch $f_G(n) = |E(G)(n)|$.

Anmerkung: Ist G propagierend (und f_G injektiv), so ist f_G (streng) monoton steigend.

Anwendung: Es gibt kein PD0L-System für $L = \{(ab^2a)^{2^n} \mid n \in \mathbb{N}\}$.

Beweis: Gäbe es solch ein System G , so wäre $E(G)(0) = ab^2a$ und $E(G)(1) = ab^2aab^2a$, d.h., $ab^2a \Rightarrow ab^2aab^2a$ wäre eine gültige Ableitung in G .

Die Regel $a \rightarrow w$ erfordert, dass w sowohl Präfix als auch Suffix von ab^2aab^2a ist (nicht-überlappend), d.h., $w = \lambda$ oder $w = a$ oder $w = ab^2a$.

Da G propagierend, scheiden die erste und die letzte Möglichkeit aus.

Mit $w = a$ gilt $b^2 \Rightarrow b^2aab^2$, was für kein $b \rightarrow v$ zutreffen kann.

Frage: D0L-System für L ?!

Wachstumsmuster bei D0L-Systemen

Beobachtung: f_G hängt nicht ab von der Abfolge der Buchstaben im Axiom oder in rechten Regelseiten; es kommt nur darauf an, wie viele Zeichen von welcher Sorte auftauchen.

Diese Information kann in einem Vektor (dem *Parikh-(Zeilen-)Vektor* des Axioms) und in einer Matrix (der *Wachstumsmatrix* der Regeln) gespeichert werden.

Wachstumsmuster bei D0L-Systemen: Ein Beispiel

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

Zugehörige 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-(Zeilen-)Vektor von abc^2 , die zweite Zeile der Parikh-Vektor von bc^2 und die dritte Zeile der Parikh-Vektor der rechten Seite der Regel $c \rightarrow c$.

Wachstumsmuster bei D0L-Systemen: Ein Beispiel (Forts.)

Welche Matrix (von Parikh-Vektoren) [bezüglich der Symbole a, b und c] beschreibt die Ableitung nach zwei Schritten ? Betrachte die Matrix

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

~> Der Parikh-Vektor des aus dem Axiom in zwei Schritten ableitbaren Wortes kann berechnet werden durch Multiplikation des Parikh-Vektors des Axioms mit M_G^2 :

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

Wachstumsmuster bei D0L-Systemen: Matrixalgebra. . .

. . . kann man einsetzen, um Wachstumsmustern auszurechnen.

~> Ausflug in Lineare Algebra.

Ist M eine $n \times n$ -Matrix (über einem kommutativen Ring), so definiert die Determinante von $x\mathbb{I}_n - M$ das *charakteristische Polynom* $p_M(x)$ von M .

Hierfür gilt das: (**Cayley-Hamilton Theorem**): $p_M(M) = 0$.

Daraus folgt unmittelbar:

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

für gewisse Zahlen c_i . ~> Rekursion für die Wachstumsmustern f_G :

$$f_G(i+n) = c_1 f_G(i+n-1) + c_2 f_G(i+n-2) + \dots + c_n f_G(i)$$

für alle $i \geq 0$. Das gestattet oft eine explizite Darstellung von f_G .

Wachstumsmuster bei D0L-Systemen: Ein Beispiel (Forts.)

Berechne M_G^3 für das obige Beispiel:

$$M_G^3 = \begin{pmatrix} 1 & 3 & 12 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{pmatrix} = c_1 \begin{pmatrix} 1 & 2 & 6 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix} + c_2 \begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} + c_3 \mathbb{I}_3,$$

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

$$\leadsto f_G(i+3) = 3f_G(i+2) - 3f_G(i+1) + f_G(i)$$

mit den Anfangswerten

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

$\leadsto f_G(i) = i^2$, denn:

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

Eine Folgerung aus dem Satz von Hamilton-Cayley

Satz: Es sei f eine D0L Wachstumsfunktion mit beliebig langen Intervallen, auf der f konstant ist, d.h. formal:

$$\forall k \exists i : f(i) = f(i + 1) = \dots = f(i + k).$$

Dann ist f *schließlich konstant*, d.h., $\exists N \exists M \forall n \geq N : f(n) = M$.

Folgerung: Ist f eine D0L Wachstumsfunktion, so gilt: $f(n) \notin \Theta(\log(n))$.

Noch ein Ausflug in die Mathematik: Funktionentheorie

Eine Funktion $F : \mathbb{C} \rightarrow \mathbb{C}$ heißt *beschränkt rational* gdw. F lässt sich darstellen als

$$F(z) = \frac{Q(z)}{P(z)} = \frac{q_0 + q_1z + \cdots + q_i z^i}{1 + p_1z + \cdots + p_j z^j} = \sum a_n z^n,$$

wobei die a_j einer der folgenden zwei Bedingungen genügen:

1. $\exists N (\forall n \geq N : a_n = 0) \wedge (\forall n < N : a_n > 0)$ oder
2. $(\forall n : a_n > 0) \wedge \exists c \forall n : \frac{a_{n+1}}{a_n} \leq c$.

Eine Wachstumsfunktion f kann man als Koeffizienten der Reihe $F(z) = \sum f(n)z^n$ auffassen. $F(z)$ kann man als (nicht notwendig stets definierte) Abbildung $F : \mathbb{C} \rightarrow \mathbb{C}$ ansehen.

Mitteilung: $F(z) = a_n z^n$ ist beschränkt rational gdw. $f(n) = a_n$ ist D0L-Wachstumsfunktion.

Literature

- G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages (3 volumes)*. Springer, 1997.
In particular: **Vol. 1, Chap. 5 & Vol. 3, Chap. 9**
- G. Rozenberg and A. Salomaa. *The Mathematical Theory of L Systems*. Academic Press, 1980.
- P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. New York: Springer, 1990.

These classical sources mostly deal with **complete** parallelism.

Many papers concerning **partial** parallelism.

[Article on this course:](#)

- H. Fernau. Parallel grammars: a phenomenology. *GRAMMARS*, vol. 6 (2003), pp. 25–87.

Phenomena

AIM: A (formal language) model of **parallel computation**

More specifically: a **grammar model**

Parallel execution of **operations**

Basic **operation** in grammars:

one **derivation step** \rightsquigarrow **parallel derivation step**

Details to come ;-)

A. Lindenmayer I

starting point: modeling algae growth

filamentous organisms are strings (of cells)

Read the masters:

A. Lindenmayer. Mathematical models for cellular interactions in development
I. Filaments with one-sided inputs. *Journal of Theoretical Biology*, 18:280–299,
1968.

A. Lindenmayer II Basic ideas of Lindenmayer's model:

1. Each cell is always in one of a finite number of **states**.
2. Cells may **interact** with their immediate (at most two) neighbors.
3. Cells may assume a new state depending on their own and their neighbors' states.
4. For an external observer, cells “process” in **parallel**;
5. these observations are undertaken after **discrete time steps**.

A. Lindenmayer III Mathematical translation of Lindenmayer's model:

1. finite-state model
2. transition function / relation takes at most three arguments
3. output of transition function / relation is again a (new) state
4. transition is applied to each cell (state) in parallel;
5. development over time modeled by repetitions of the application of the transition function / relation

A. Lindenmayer IV Is this all of Lindenmayer's model? NO

1. cells may multiply / divide

2. cells may die



grammatical model rather than automaton model (**cellular automata**)

if a cell is then a character of a string, then **dying** means becoming the **empty word**

DIL Systeme

Lindenmayer-Systeme mit Interaktion zwischen Nachbarn (I), die aber deterministisch arbeiten.

~> Auch für solche Systeme ist der Begriff der Wachstumsfunktion definierbar.

If there are not $i > 0$ symbols to the left (e. g., when considering the leftmost symbol), a special symbol # is considered as being read. Similarly, “missing” right neighbours are handled.

Given a DIL system $G = (\Sigma, h, \omega)$, the *DIL growth function*

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

gives the length of the word derived after n steps.

Lemma: No DIL growth function grows faster than exponential.

Example: Consider the DIL system with axiom ad and the rules:

	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

The derivation proceeds as follows:

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

It can be observed that this system, also known as *Gabor's sloth*, grows at a logarithmic scale.

What **kinds of functions** can be realized as DIL growth functions?

1. exponential growth: consider the D0L system with rule $a \rightarrow a^2$;
2. polynomial growth: $a \rightarrow ab, b \rightarrow b$ generate, starting with a , ab^n ;
3. logarithmic growth: see Gabor's sloth.

Folgerung: Es gibt kein D0L-System, dessen Wachstumsfunktion der von Gabor's Faultier entspricht.

Folgerungen für Sprachklassen

Es gibt DIL-Sprachen, die keine D0L-Sprachen sind.

Es gibt D0L-Sprachen, die keine PD0L-Sprachen sind.

Beobachte: Diese Folgerungen wurden durch sehr unterschiedliche mathematische Überlegungen erzielt !

A. Lindenmayer V Extensions of Lindenmayer's model

A. Lindenmayer. Mathematical models for cellular interactions in development II. Simple and **branching filaments** with two-sided inputs. *Journal of Theoretical Biology*, 18:300–315, 1968.

A. Lindenmayer. *Adding **continuous components** to L-systems*, IN: volume 15 of *LNCS*, pages 53–68. Berlin: Springer, 1974.

A. Lindenmayer VI Modelling higher organisms

- growth only occurs near the boundary
- L systems with apical growth
- cellular automata (J von Neumann 1950..)
- (array grammars) [mostly viewed in sequential variants]

Literature

A. Burks. *The Theory of Self-Reproducing Automata* contains a chapter of J. v. Neumann (a reprint of earlier work?). The University of Illinois Press, 1966.

V. Aladyev. τ_n -grammars and their generated languages (in Russian). *Transactions of the Academy of Sciences of Estonia; Eest NSV Teaduste Akadeemia toimetised / Biologiline seeria*, 23(1):67–87, 1974.

V. Aladyev. Operations on languages generated by τ_n -grammars (in Russian). *Commentationes Mathematicae Universitatis Carolinae*, 15:211–220, 1974.

V. Aladyev. On the equivalence of τ_m -grammars and $Sb(n)$ -grammars (in Russian). *Commentationes Mathematicae Universitatis Carolinae*, 15:717–726, 1974.

N. Nirmal and K. Krithivasan. Filamentous systems with apical growth. *International Journal of Computer Mathematics*, 12:203–215, 1983.

A. Lindenmayer VII Modelling higher organisms *more realistically*

- 2-3 dimensions:

A. Paz. *Multidimensional parallel rewriting systems*, IN: Automata, Languages, Development, pages 509–515. North-Holland, 1976.

- *computer graphics* rendering: Prusinkiewicz

A shift of interests I Mathematics / Computer Science influence

Mathematical beauty

~> study of L systems **without interaction**

A. Lindenmayer. Developmental systems without cellular interactions, their languages and grammars. *Journal of Theoretical Biology*, 30:455–484, 1971.

A shift of interests II Biology influence

interest in the **development of organisms** (over time)



- study of **growth patterns**: PhD theses of P. G. Doucet and of P. M. B. Vitányi, the latter one with the title: Lindenmayer systems: structure, languages, and growth functions.
- **partial parallel derivations**, **ecogrammar systems** might avoid **exponential / polynomial growth**

Partial parallelism Literature overview (partially chronological)

- multihead finite automata / TMs
- finite index restriction \rightsquigarrow **absolutely parallel grammars**

V. Rajlich. Absolutely parallel grammars and two-way deterministic finite state transducers. *Journal of Computer and System Sciences*, 6:324–342, 1972.

- these can be generalized: **scattered context grammars**

S. Greibach and J. Hopcroft. Scattered context grammars. *Journal of Computer and System Sciences*, 3:233–247, 1969.

J. Gonczarowski and M. K. Warmuth. Scattered versus context-sensitive rewriting. *Acta Informatica*, 27:81–95, 1989.

many papers of A. Meduna

- Simple matrix grammars

O. Ibarra. Simple matrix languages. *Information and Control (now Information and Computation)*, 17:359–394, 1970.

A. Pascu and Gh. Păun. On simple matrix grammars. *Bull[etin] Math. de la Soc. Sci. Math. [de la R. S.] de Roumanie*, 20:333–340, 1976.

- n -parallel automata and grammars

R. D. Rosebrugh and D. Wood.

A characterization theorem for n -parallel right linear languages. *Journal of Computer and System Sciences*, 7:579–582, 1973.

Image theorems for simple matrix languages and n -parallel languages. *Mathematical Systems Theory*, 8:150–155, 1974.

Restricted parallelism and right linear grammars. *Utilitas Mathematica*, 7:151–186, 1975.

- **Indian parallelism / Bharat systems**

R. Siromoney and K. Krithivasan. Parallel context-free languages. *Information and Control (now Information and Computation)*, 24:155–162, 1974.

M. Kudlek. Languages defined by Indian parallel systems. In G. Rozenberg and A. Salomaa, editors, *The Book of L*, pages 233–244. Berlin: Springer, 1985.

- **(uniformly) limited Lindenmayer systems**

D. Wätjen. k -limited 0L systems and languages. *J. Inf. Process. Cybern. EIK (formerly Elektron. Inf.verarb. Kybern.)*, 24(6):267–285, 1988.

K. Salomaa. Hierarchy of k -context-free languages. *International Journal of Computer Mathematics*, 26:69–90,193–205, 1989.

D. Wätjen. On k -uniformly-limited T0L systems and languages. *J. Inf. Process. Cybern. EIK (formerly Elektron. Inf.verarb. Kybern.)*, 26(4):229–238, 1990.

- parallel communicating grammar systems

\rightsquigarrow Lecture on grammar systems!

- (Pattern languages)

D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.

- Concepts can be combined:

- Russian parallel grammars

M. K. Levitina. On some grammars with rules of global replacement (in Russian). *Scientific-technical information (Nauchno-tehnicheskaya informacii)*, Series 2, (3):32–36, 1972.

- PCGS with partial parallelism

D. Wätjen. Parallel communicating limited and uniformly limited 0L systems. *Theoretical Computer Science*, 255(1–2):163–191, 2001.

Features to be modeled

- (partial) parallelism (see above)
- **synchronization**
 - (limited) L systems
 - PCGS
 - **trace languages**

V. Diekert and G. Rozenberg, editors. *Book of Traces*. World Scientific, Singapore, 1995.

- communication
 - neighborhood patterns in cellular automata
 - PCGS modes
- Communicating distributed grammar systems CDGS (Csuhaaj-Varjú)

Parallel side tracks

- parallelism in operators
(Kudlek / Mateescu; traces)
- picture processing
 - collage grammars (Kreowski/Habel/Drewes)
 - array grammars (Freund/Wang)
 - iterated function systems
- term rewriting
- “parallelism degrees” in context-free grammars (Brandenburg/Reinhardt)

Phenomena translation (The Lindenmayer Alphabet)

P	$A \rightarrow \lambda$	erasing rule	cell death
(G)	$A \rightarrow A$	propagating chain rule	cells don't die
I	$A \rightarrow BC$	growing rule	state change
J	$XAY \rightarrow XBY$	context-sensing	cell division
T		(Finnish: J akautua)	interaction
A		Tables	fragmentation
C		Adult mechanism	environment conditions
		Codings	only grown-ups count
			non-observable features

Further Phenomena

Signals play an eminent role in CAs.

Determinism \rightsquigarrow DL systems

(a special terminal alphabet \rightsquigarrow **E**xtension)