Lindenmayer-Systeme

Zusammenfassung des Vortrags im Proseminar Theoretische Informatik

Benjamin Aram Berendsohn

11. November 2014

1 Geschichte

Lindenmayer-Systeme, kurz *L-Systeme*, wurden zuerst 1968 von dem Biologen Aristid Lindenmayer beschrieben, um bestimmte biologische Wachstumsvorgänge zu modellieren. Sie ähneln den Chomsky-Grammatiken, allerdings werden Produktionen in Lindenmayer-Systemen parallel angewendet und es existieren keine Terminalsymbole. Dies trägt der Tatsache Rechnung, dass etwa Zellteilung an mehreren Stellen im Organismus gleichzeitig stattfindet.

2 D0L-Systeme

Ein D0L-System ist definiert als Tripel (Σ, P, w_0), wobei Σ ein Alphabet, P eine endliche Menge von Tupeln in $\Sigma \times \Sigma^*$ (*Produktionen*) und $w_0 \in \Sigma^*$ das *Axiom* ist. Produktionen werden statt (σ, w) auch $\sigma \to w$ geschrieben. Jedes Zeichen in Σ ist die linke Seite genau einer Produktion.

Im Gegensatz zur Ableitung bei formalen Sprachen, die in der Anwendung einer Produktion auf ein beliebiges Zeichen oder Teilwort besteht, werden bei der Ableitung in D0L-Systemen alle Zeichen im Wort durch die rechte Seite der jeweiligen Produktion ersetzt. Dass ein Wort w_2 durch die Ableitung eines Wortes w_1 entsteht, wird als w_1 *erzeugt direkt* w_2 bezeichnet und mit $w_1 \Rightarrow w_2$ abgekürzt.

Die Folge, die entsteht, wenn das Axiom eines D0L-Systems G wiederholt abgeleitet wird, wird als *erzeugte Folge* $E(G)$ bezeichnet.

Die *erzeugte Sprache* $L(G)$ eines 0L-Systems G ist die Menge aller Wörter, die in beliebig vielen Schritten vom Axiom abgeleitet werden können: $L(G) = \{w | w_0 \Rightarrow^* w\}$.

Ein D0L-System heißt *wachsend* oder *PD0L-System*, wenn keine Produktion das leere Wort als rechte Seite hat.

2.1 Wachstumsfunktionen

Die *Wachstumsfunktion* f_G eines D0L-Systems G beschreibt die Länge des Wortes an einer bestimmten Stelle der erzeugten Folge: $f_G(n) = |E(G)_n|$.

Im Folgenden soll eine Methode gezeigt werden, die Wachstumsfunktion eines L-Systems zu finden.

Sei $G = (\{a, b\}, (a \to b, b \to ab), a)$. Dieses D0L-System ist eine Vereinfachung eines von Lindenmayer entworfenem Systems, dass das Wachstum des Fadenförmigen Bakteriums *Anabaena catenula* modelliert. b sind dabei längere, a kürzere Segmente.

Sei $f_\sigma(n)$ definiert als die Häufigkeit des Zeichens σ in dem n-ten Element von $E(G)$. b wird jeweils genau einmal von a und b erzeugt, daher gilt

$$f_b(n) = f_a(n-1) + f_b(n-1)$$

und analog

$$f_a(n) = f_b(n-1)$$

Die Wachstumsfunkion zählt alle Zeichen:

$$\begin{aligned} f_G(n+2) &= f_a(n+2) + f_b(n+2) = f_a(n+1) + 2f_b(n+1) \\ &= f_G(n+1) + f_b(n+1) = f_G(n+1) + f_a(n) + f_b(n) \\ &= f_G(n+1) + f_G(n) \end{aligned}$$

G wächst also wie die Fibonacci-Folge.

3 0L-Systeme

0L-Systeme erlauben, im Gegensatz zu D0L-Systemen, auch mehrere Produktionen für jedes Zeichen, von denen eine beliebig gewählt werden kann. Eine nichtdeterministisches 0L-System erzeugt keine Folge mehr, sondern einen Baum, da jedes Wort mehrere Wörter direkt erzeugen kann. Die erzeugte Sprache eines 0L-Systems ist genau wie die eines D0L-Systems definiert.

3.1 Einordnung in die Chomsky-Hierarchie

Die endliche Sprache $\{a, aa\}$ lässt sich durch kein 0L-System darstellen. Anders herum lässt sich die Sprache $\{a^{2^n} | n \in \mathbb{N}_0\}$ sehr leicht als D0L-System darstellen, ist aber nicht kontextfrei.

4 Turtle-Interpretation von Wörtern

Turtle-Grafiken basieren auf der Idee, einer so genannten „Schildkröte“ Anweisungen zu geben, nach denen diese sich auf der Zeichenfläche (oder im Raum) bewegt und dabei eine Grafik zeichnet.

Der *Zustand* der Schildkröte wird durch einen Ortsvektor und (im zweidimensionalen Fall) einem Ausrichtungswinkel dargestellt. Gegeben ist ein Startzustand. Die Schildkröte kann (unter anderem) folgende Anweisungen ausführen:

$F(d)$: Vorwärtsbewegung um die Länge d. Eine Linie wird zwischen Anfangs- und Endpunkt gezeichnet.

$+(\delta)$: Drehung nach links um den Winkel δ.

In der Regel werden d und δ vorher definiert und in einem Anweisungsstring F als $F(d)$, $+$ als $+(\delta)$ und $-$ als $+(-\delta)$ interpretiert. Oft wird auch d gar nicht definiert und die erzeugte Grafik nach Wunsch skaliert (so wird in den folgenden Beispielen vorgegangen). Genauso ist der Startpunkt und Startausrichtung der Schildkröte in aller Regel unwichtig.

Turtle-Grafiken werden etwa in der Lernprogrammiersprache LOGO verwendet. Von Przemyslaw Prusinkiewicz stammt die Idee, von Lindenmayer-Systemen erzeugte Wörter als Turtle-Grafiken zu interpretieren. So konnten Pflanzen auch grafisch modelliert werden.

4.1 Fraktale

Lindenmayer-Systeme in Verbindung mit Turtle-Grafiken eignen sich gut, bestimmte Figuren zu erzeugen. Ein Beispiel ist die Koch-Kurve, die von folgendem Lindenmayer-System erzeugt wird:

$\delta = 60°$

Axiom: F

$F \to F + F - - F + F$

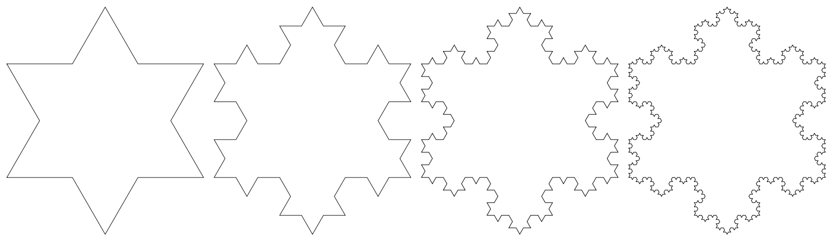

Abbildung 1: Die ersten 4 Iterationen der Koch-Schneeflocke, dreier als Dreieck angeordneter Koch-Kurven

Die Koch-Kurve ist ein *Fraktal*. Fraktale sind Geometrische Objekte (genauer: Mengen von Punkten), die eine gebrochene *Hausdorff-Dimension* aufweisen. Diese ist wie folgt definiert: Sei $N(\varepsilon)$ die Anzahl an Bällen mit Radius ε, die benötigt werden, um das gegebene Objekt vollständig abzudecken. Die Hausdorff-Dimension D ist die Potenz, mit der $N(\varepsilon)$ bei gegen 0 strebendem ε wächst:

$$D = \lim_{\varepsilon \to 0} - \frac{log(N(\varepsilon))}{log(\varepsilon)}$$

Bei rekursiv definierten Objekten wie der Koch-Kurve ist die Hausdorff-Dimension oft einfach zu bestimmen: Die Koch-Kurve besteht selbst aus 4 um $\frac{1}{3}$ verkleinerten Koch-Kurven. In jedem Schritt, der ε um drei kleiner werden lässt, wird die N vervierfacht. Man kann zeigen, dass daraus $D = \frac{log(4)}{log(3)}$ folgt; D ist keine ganze Zahl.

Einfache geometrische Objekte wie Linien und Quadrate haben ganze Hausdorff-Dimensionen.

4.2 Geklammerte L-Systeme

Die beschriebenen Befehle sind geeignet, um Linienzüge zu zeichnen, eignen sich aber nicht, um Verzweigungen, wie sie bei Pflanzen in der Regel vorkommen, zu beschreiben.

Dazu wird die Schildkröte um einen Kellerspeicher (Stack) für Zustände und der Befehlssatz wie folgt erweitert:

[: Lege den aktuellen Status der Turtle auf den Stack.

]: Nehme einen Status vom Stack und ersetze den aktuellen mit jenem.

So erweiterte L-Systeme werden *geklammerte L-Systeme* genannt.

4.3 Sonstige Turtle-Erweiterungen

Um Organismen grafisch zu modellieren, werden in der Regel drei Dimensionen benötigt. Dafür wird der Ausrichtungswinkel im Zustand der Turtle durch einen Richtungsvektor ersetzt, und Befehle zum Drehen nach oben und unten und Rollen zur Seite hinzugefügt.

Eine weitere wichtige Erweiterung ist die Fähigkeit, ausgefüllte Polygone zu zeichnen. Dabei werden Befehle zum Starten und Abschließen von Polygonen sowie zum Setzen von Wegpunkten hinzugefügt.

Außerdem kann ein *Tropismusvektor* definiert werden. Alle Linien werden etwas in Richtung dieses Vektors gedreht. So kann man einfach Schwerkraft oder andere Umwelteinflüsse simulieren.

5 Weitere Variationen von L-Systemen

Es existieren noch viele Erweiterungen der einfachen 0L-Systeme, von denen einige hier kurz beschrieben werden sollen.

Stochastische L-Systeme sind nichtdeterministische 0L-Systeme, bei denen jeder Produktion eine Wahrscheinlichkeit zugeordnet wird. Bei der Generation von Grafiken werden Produktionen mit der gegebenen Wahrscheinlichkeitsverteilung zufällig angewandt. Dies erlaubt zum Beispiel Variationen in sonst sehr eintönigen generierten Landschaften.

IL-Systeme sind kontextsensitive L-Systeme. In IL-Systemen hat jeden Produktion einen linken Kontext der Länge k und einen rechten Kontext der Länge l, jeweils ein Wort über $\Sigma \cup \{\_\}$, wobei $\_$ für das Leerzeichen steht. Produktionen ersetzen wie bei 0L-Systemen ein einzelnes Zeichen durch ein Wort, werden aber nur angewandt, wenn der Kontext des Zeichens mit dem der Produktion übereinstimmt. IL-Systeme für bestimmte k und l werden (k, l)L-Systeme genannt. $(0, 1)$L- und $(1, 0)$L-Systeme werden *1L-Systeme* genannt, $(1, 1)$L-Systeme *2L-Systeme*. $(0, 0)$L-Systeme entsprechen offensichtlich den 0L-Systemen.

T0L-Systeme (bzw. TIL-Systeme) sind L-Systeme mit mehreren Produktionsmengen. Diese werden nach bestimmten Regeln ausgetauscht, etwa nach einer festgelegten Anzahl von Schritten. Dies entspricht dem Verhalten mancher Pflanzen, etwa nach einer bestimmten Anzahl von Blättern eine Blüte zu erzeugen.

Parametrisierte L-Systeme erlauben das Hinzufügen von numerischen Parametern zu Zeichen in Axiom und Produktionen. Diese Parameter können von Produktion geprüft und weiterverarbeitet werden; und auerdem die Turtle-Interpretation beeinflussen (z.B. die Länge einer gezeichneten Linie bestimmen). Nachfolgend ein Beispiel:

$$F(s, t, c) : t = 1 \land s \geq 6 \rightarrow F(\frac{3s}{2}, 2, c)f(1)F(\frac{s}{3}, 1, c)$$

Die Boolesche Formel nach dem Doppelpunkt bestimmt, unter welchen Umständen die Produktion angewandt wird. Man sieht außerdem, dass die Parameter in der rechten Seite ebenfalls vorkommen.

Parametrisierte L-Systeme spielen eine große Rolle in der Generation von Pflanzen, da sie nicht-diskrete Modellierung erlauben.

Literatur

[1] P. Prusinkiewicz, A. Lindemayer: *The Algorithmic Beauty of Plants*. Springer-Verlag, New York 1990.

[2] G. Rozenberg, A. Salomaa: *The Mathematical Theory of L Systems*. Academic Press, New York 1980.

[3] A. Salomaa: *Formale Sprachen*. Springer-Verlag, Berlin 1978.

[4] G. T. Hermann, G. Rozenberg: *Developmental Systems and Languages*. North-Holland Publishing Company, Amsterdam 1975.

[5] 1997 Computer Graphics Achievement Award. http://old.siggraph.org/awards/1997/AchievementAward.html (Abgerufen am 9.11.2014, 12:19)

[6] Hausdorff dimension | planetmath.org. http://planetmath.org/HausdorffDimension (Abgerufen am 9.11.2014, 12:22)