Obliczenia inspirowane Naturą

Wykład 05 – Biologia i gramatyka

Jarosław Miszczak

IITiS PAN Gliwice

07/04/2016

Na poprzednim wykładzie

1. Nieformalne określenie fraktali.
2. Wymiar pudełkowy/fraktalny.
3. Definicja fraktali.

1. L-Systemy
   - Motywacja biologiczna
   - Systemy przepisywania ciągów
   - Przykłady
   - Interpretacja graficzna
   - Warianty L-systemów
2. Gramatyki
   - Systemy półthueowskie
   - Hierarchia Chomskiego
   - Postać Backusa-Naura
   - Wyrażenia regularne

L-systemy

Motywacja biologiczna

L-systemy czyli systemy Lindenmayera.

- 1968, Aristid Lindenmayer – wykorzystanie do opisu rozwoju alg i innych prostych organizmów wielokomórkowych.
- L-systemy to przykład systemu przepisywania ciągów (ang. *string rewriting system*).
- Transformują one zadany ciąg wejściowy za pomocą gramatyki czyli zespołu reguł.
- Reguły są wywoływane rekurencyjnie co prowadzi do pojawienia się struktur samopodobnych.

L-systemy

Co to jest przepisywanie?

Przepisywanie

Metoda tworzenie nowych obiektów poprzez zastępowanie ich części.

Abstrakcyjny systemy przepisywania

Zbiór A wraz z binarną relacją, $\rightarrow \subset A \times A$, nazywaną redukcją (ang. *reduction relation* lub *rewrite relation*).

System przepisywania nie określa algorytmu, a jedynie reguły przekształcania obiektów – podobnie jak deklaratywne języki programowania.

L-systemy

Systemy przepisywanie ciągów

Systemy przepisywanie ciągów

Bazuje na strukturze wolnego monoidu, czyli możliwości operowania na skończonych ciągach elementów zbioru.

Monoid

Zbiór A z elementem jednostkowym i z operacją binarną, która jest łączna, $((x \cdot y) \cdot z = x \cdot (y \cdot z))$. Inaczej: półgrupa z jednością.

Wolny monoid

Zbiór wszystkich skończonych ciągów elementów z monoidu A o długości zero lub więcej.

L-systemy
Systemy przepisywanie ciągów

System przepisywania ciągów to para $(A, \rightarrow)$, gdzie

- A jest alfabetem
- $\rightarrow$ to relacja binarna na $A^\star$, $\rightarrow \subset A^\star \times A^\star$

W interesującym nas przypadku identyczność do napis pusty oznaczany jako ϵ.

L-systemy
Gwiazdka Kleene'a

Dla zadanego zbioru A definiujemy ciąg zbiorów

- $A_0 = \{\epsilon\}$
- $A_1 = A$
- $A_n = \{\alpha a : \alpha \in A_{n-1}, a \in A\}$

Gwiazdka Kleene'a

$$A^\star = \bigcup_{n \in \mathbf{N}} A_n$$

Plus Kleene'a

$$A^+ = \bigcup_{n \in \mathbf{N} \setminus \{0\}} A_n$$

L-systemy
Definicja

L-system

L-system jest zdefiniowany jako trójka (A, α, P), gdzie

- A jest alfabetem
- $\alpha \in A^\star$ to **aksjomat** czyli symbol startowy
- P – relacja binarna, tzw. wyprowadzenie, $P : A \to A^\star$

L-systemy
Zasada działania

L-systemy polegają na **równoległym** wykonaniu reguł zgodnie z poniższymi zasadami:

- reguły są stosowane iteracyjnie;
- w każdej iteracji są zastosowane wszystkie możliwe reguły;
- stałe to symbole z alfabetu, które nie pojawiają się po lewej stronie żadnej reguły.

L-systemy

Własności

Systemy D0L

L-system nazywany jest **bezkontekstowym** gdy każda reguła produkcji stosuje się tylko do pojedynczego symbolu.

Jeśli reguła produkcji zależy nie tylko od pojedynczego symbolu, ale także od symboli sąsiednich, to taki L-system nazywany jest **kontekstowym**.

Systemy stochastyczne

Jeśli dla danego symbolu reguły produkcji są przypisane z pewnym prawdopodobieństwem, to taki L-system nazywa się stochastycznym.

L-Systemy

Oryginalny przykład Lindenmayera

Oryginalnie Lindenmayer wykorzystał L-system do opisu wzrostu alg. Proces ten modelował za pomocą następującego systemu:

- **zmienne**: a, b
- **aksjomat**: a
- **reguły** (**produkcje**): $(a \rightarrow ab)$, $(b \rightarrow a)$

L-Systemy

Oryginalny przykład Lindenmayera

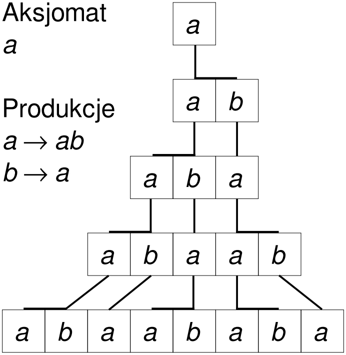
Aksjomat

a

Produkcje

$a \rightarrow ab$

$b \rightarrow a$



L-Systemy
Oryginalny przykład Lindenmayera

- $n = 0 : a$
- $n = 1 : ab$
- $n = 2 : aba$
- $n = 3 : abaab$
- $n = 4 : abaababa$
- $n = 5 : abaababaabaab$
- $n = 6 : abaababaabaababaababa$
- $n = 7 :$ abaababaabaababaababaabaababaabaab

L-Systemy

Oryginalny przykład Lindenmayera

Otrzymujemy w ten sposób **słowa Fibonacciego** z pominięciem pierwszego elementu.

$$F_n := \begin{cases} b & \text{dla } n = 1; \\ a & \text{dla } n = 2; \\ F_{n-1} \cdot F_{n-2} & \text{dla } n > 2. \end{cases}$$

Gdzie '$\cdot$' oznacza **konkatenację** czyli łączenie ze sobą wyrażeń (operator '$<>$' w *Mathematica*, '$+$' w Java/C$++$/Python, '.' w Perl/PHP)

L-Systemy
Interpretacja graficzna

Ciąg (napis) wygenerowany przez L-system może być łatwo zinterpretowany jako ciąg komend.

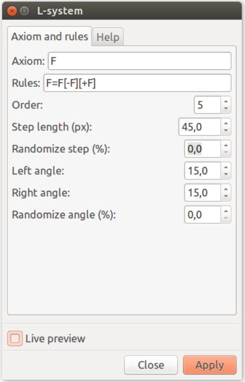
- F – rysuj do przodu (o zadaną długość)
- f – przesuń się do przodu (o zadaną długość)
- $+$/- – wykonaj obrót w prawo/lewo o zadany kąt.

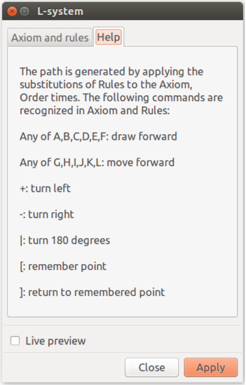
W ten sposób ciąg wygenerowany przez L-system jest interpretowany jako program do rysowania. Naśladuje to metodę programowania znaną z języka Logo.

Interpretacja graficzna L-systemów

Generowanie roślin

Program Inkscape udostępnia prosty interfejs do generowania grafiki wektorowej za pomocą L-systemów (Extensions → Render → L-System).





Interpretacja graficzna L-systemów

Smok Heighwaya

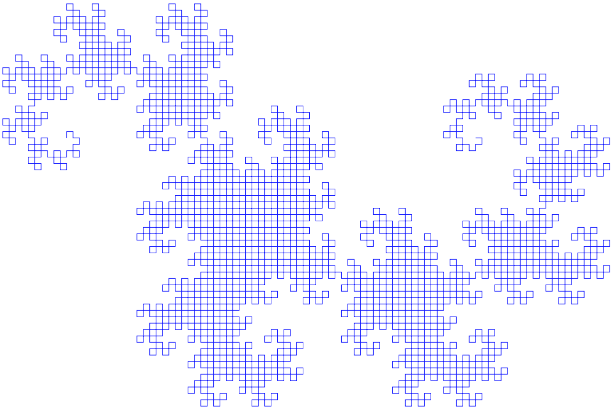
Smok Heighwaya (ang. *dragon curve*) jest określona poprzez następujący L-system:

- **zmienne**: X, Y
- **stałe**: F, $+$, -
- **start**: FX
- **reguły**: $\mathrm{X} \rightarrow \mathrm{X+YF+}$, $\mathrm{Y} \rightarrow \mathrm{-FX-Y}$
- **kąt**: 90°

Tutaj, F oznacza "rysuj odcinek", '-' oznacza "obrót w lewo", a '$+$' oznacza "obrót w prawo". W tym wypadku symbole X i Y są wykorzystywane do kontrolowania ewolucji systemu i nie są interpretowane jako polecenia rysowania.

Interpretacja graficzna L-systemów

Smok Heighwaya



Dwunasta iteracja Smok Heighwaya w programie Inkscape

Interpretacja graficzna L-systemów

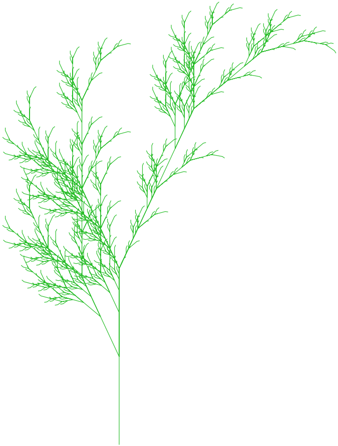
Generowanie roślin

- **zmienne**: X, F
- **stałe**: $+$, -, $[$, $]$
- **aksjomat**: X
- **reguły**: $\mathrm{X} \to \mathrm{F\text{-}[[X]+X]+F[+FX]\text{-}X}$, $\mathrm{F} \to \mathrm{FF}$

Tutaj F oznacza "rysuj do przodu", a X nie ma interpretacji graficznej.
Symbole - "obrót w lewo" i $+$ "obrót w prawo" o kąt 25°. Symbole $[$ oznacza "wrzuć na stos", a $]$ pozycję i kąt.

Interpretacja graficzna L-systemów

Generowanie roślin



Interpretacja graficzna L-systemów

Generowanie roślin

Interpretacja graficzna może dopuszczać elementy losowe. W najprostszym przypadku możemy uwzględnić:

- losowość kąta
- losowość długości przesunięcia

Ograniczeniem są tu tylko możliwości języka w którym interpretujemy L-system.

Interpretacja graficzna L-systemów

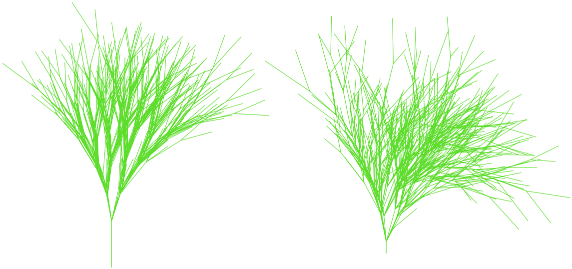
Generowanie roślin

Dodanie losowości do kątów i długości kroku powoduje powstanie bardziej naturalnych roślin.

Interpretacja graficzna L-systemów

Generowanie roślin

Dodanie losowości do kątów i długości kroku powoduje powstanie bardziej naturalnych roślin.



Nie jest to L-system probailistyczny – losowość występuje tylko w interpretacji graficznej.

L-systemy

Klasyfikacja L-systemów

- D0L-system – deterministyczny, bezkontekstowy (0 oznacza długość kontekstu);
- D1L-system – deterministyczny, kontekstowy;
- 0L-system – stochastyczny, bezkontekstowy;
- 1L-system – stochastyczny, kontekstowy,
- (1,1)L-system (albo 2L) – stochastyczne, wrażliwy na kontekst lewo- i prawostronny;

L-Systemy

Reguły probabilistyczne

W L-systemach deterministycznych każdemu symbolowi odpowiadała dokładnie jedna reguła (produkcja). W systemie probabilistycznym:

- regułom przypisane są prawdopodobieństwa wykorzystanie ich w trakcie wykonania;
- przykładowo
  - reguła deterministyczna: $0 \rightarrow 1[0]0$
  - reguła probabilistyczna: ((0.5, $0 \rightarrow 1[0]0$), (0.5, $0 \rightarrow 0$)) – dwie reguły, każda ma przypisane prawdopodobieństwo zastosowania

L-Systemy

Reguły kontekstowe

W systemach D0L reguły są jednoznaczne określone dla symbolu.

Systemy D1L

W systemach D1L reguły są zależne od symboli z którymi sąsiaduje przekształcany symbol

- $\mathrm{b\ a\ c} \rightarrow \mathrm{aa}$ – zamień a na aa tylko jeżeli jest w otoczeniu b i c
- jeden symbol może mieć przypisanych wiele reguł zależnych od jego sąsiedztwa.

L-Systemy
Reguły parametryczne

Systemy parametryczne pozwalają na przekazywanie informacji wykorzystywanej do interpretacji graficznej lub w regułach.

- załóżmy, że symbol ma dwa parametry, $\mathrm{a(x,y)}$ – jest ona wówczas nazywana modułem;
- reguła może wykorzystywać informacje o parametrach, np. $\mathrm{a(x,y) : x == 0 \rightarrow a(1,\ y{+}1)b(2,3)}$ zostanie wykonana tylko jeżeli pierwszy parametr jest równy zero.

L-Systemy
Reguły parametryczne

- Systemy parametryczne pozwalają na określenie długości segmentu i kąta za pomocą gramatyki, a nie interpretacji graficznej.
- Jeżeli moduł przechowuje informacje o wieku, gramatyk może uwzględniać starzenie się segmentów.

Gramatyki

Systemy półthueowskie

1914, Axel Thue – systematyczny opis systemu przepisywania ciągów.

- L-systemy są przykładem ogólnego systemu przepisywania.
- Systemy przepisywania ciągów są określane jako systemy półthueowskie (ang. *semi-Thue grammar*).
- Jako formalizm, systemu przepisywania ciągów są **zupełne w sensie Turinga** – mogą symulować uniwersalną maszynę Turinga.

Gramatyki

Zupełność w sensie Turinga

- System zupełny w sensie Turinga (ang. *Turing-complete*) może wykonać dowolny algorytm.
- Symulacja nie musi być wydajna.
- Zupełność jest jedynie stwierdzeniem równoważności, nie daje natomiast opisu symulacji.
- Większość języków programowania jest zupełna w sensie Turinga.
- Języki które nie są zupełnie to: SQL, HTML i wyrażenia regularne.

Gramatyki
Hierarchia Chomskiego

1956, Noam Chomsky – hierarchia języków i gramatyk formalnych.

- **Typ 0** rekurencyjnie przeliczalny
- **Typ 1** kontekstowe
- **Typ 2** bezkontekstowe
- **Typ 3** regularne

N. Chomsky, *Three models for the description of language*, IRE Trans. Inf. Theory, 2 (3), pp. 113-124 (1956).

Gramatyki
Hierarchia Chomskiego

- regularne – reguły powstają przez doklejanie kolejnych symboli;
- bezkontekstowe – reguły nie zależą od otoczenia symbolu;
- kontekstowe – reguły zależą od otoczenia symbolu;
- rekurencyjnie przeliczalny – brak ograniczeń na zbiór reguł.

Gramatyki
Gramatyki formalna

- **Gramatyka formalna** – metoda opisu języka formalnego rozumianego jako podzbiór zbioru wszystkich słów skończonej długości nad danym alfabetem.
- **Symbol nieterminalny** – symbol który jest definiowany przez reguły gramatyki.
- **Symbol terminalny** – symbole które nie mogą być zmieniona za pomocą reguł gramatyki.

Jeden język może być generowany przez wiele gramatyk.

Gramatyki
Postać Backusa-Naura

1963, Peter Naur, raport języka ALGOL 60.

- BNF (ang. *Backus Normal Form* lub ang. *Backus-Naur Form*) to sposób zapisu reguł wyprowadzenia w gramatykach bezkontekstowych.
- Jest to **metajęzyk** – język opisu języków programowania.

Gramatyki
Postać Backusa-Naura

Postać Backusa-Naura jest wykorzystywana do opisu języków bezkontekstowych – wszystkie reguły (produkcje) są postaci:
A $\rightarrow \alpha$
Kilka przykładów:

- Liczby całkowite:
  <integer>::= <digit>| <integer><digit>
- Składnia funkcji w języku Python:
  <funcdef>::== 'def' NAME <parameters>':' <suite>

Symbole które nigdy nie pojawiają się po lewej stronie (np. NAME) to symbole terminalne. Są to symbole stałe dla gramatyki – w językach kontekstowych mogą pojawić się po lewej stronie.

Gramatyki

Gramatyki kontekstowe

Gramatyka to czwórka (X, T, P, s) gdzie

- X to symbole nieterminalne,
- T to wyróżnione symbole terminalne,
- P to zbiór reguł (produkcji),
- s to symbol początkowy.

W przypadku gramatyk kontekstowych reguły są postaci

$$\alpha A \beta \to \alpha \gamma \beta$$

gdzie $\alpha, \gamma, \beta \in$

Gramatyki

Wyrażenia regularne

1950s, Stephen Kleene – formalny opis języków regularnych.

- Wyrażenia regularne są wbudowane w większość języków programowania.
- Są szczególnie przydatne do analizy/formatowania/edycji plików tekstowych.
- Są wykorzystywane do analizy leksykalnej w parserach języków programowania.

Gramatyki

Wyrażenia regularne

Składnia wyrażeń regularnych:

- za wyjątkiem znaków specjalnych, każdy znak oznacza sam siebie;
- alternatywa: $|$;
- grupowanie: $(,)$;
- wyrażenie e występuje 0 lub 1 raz: e?;
- 0 lub więcej wystąpień e: $e*$ (gwiazdka Kleene);
- 0 lub więcej wystąpień e: $e+$ (plus Kleene);

Gramatyki

Wyrażenia regularne

Przykłady wyrażeń regularnych

- Co najmniej jedno a lub b: $(\mathrm{a}|\mathrm{b})+$
- Liczby rzeczywiste: $[+\text{-}]?[0\text{-}9]+(.[0\text{-}9]+)?.$