Obliczenia inspirowane Naturą

Wykład 05 – Gramatyki formalne (uzupełneinie Wykładu 04)

Jarosław Miszczak

IITiS PAN Gliwice

20/10/2016

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

$$\alpha A \beta \rightarrow \alpha \gamma \beta$$

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);
- 1 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: $[+\text{-}]?[0\text{-}9]+(.[0\text{-}9]+)?$.