

Stefan Sokołowski

JĘZYKI FORMALNE I METODY KOMPILACJI

wykład 5

Inst. Informatyki Stosowanej, ANS
Elbląg, 2024/2025

Wykład 5, str. 1

Notacja beznawiasowa Łukasiewicza

Notacja Łukasiewicza (w świecie nazywana *Polish*):

Wyrażenia (arytmetyczne, logiczne, itp.) można zapisywać **bez nawiasów** w postaci

$\text{operator } \arg_1 \arg_2 \dots \arg_n$

Odwrotna notacja Łukasiewicza (*Reverse Polish*):

$\arg_1 \arg_2 \dots \arg_n \text{ operator}$

stosowana w informatyce.



Jan Łukasiewicz
1878–1956

Jak to czytać?

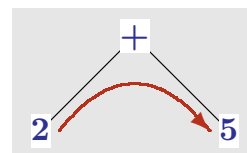
(na **czzerwono** pod operatorem — liczba argumentów czyli „arność”)

Łukasiewicza	odwrotna	czyli	klasycznie
$+ \underset{2}{2} 5$	$2 5 \underset{2}{+}$		$2 + 5$
$* \underset{2}{+} \underset{2}{2} 5 - \underset{2}{3} 1$	$2 5 \underset{2}{+} 3 1 - \underset{2}{2} *$		$(2 + 5) * (3 - 1)$
$\text{if} \underset{3}{>} x \underset{2}{0} P Q$	$x \underset{2}{0} > P Q \underset{3}{\text{if}}$		$\text{if}(x > 0) P \text{ else } Q$

Wyrażenia — drzewa (nie całkiem drz. wyvodu).

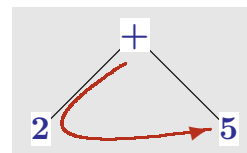
Różne **notacje** — sposoby chodzenia po drzewach.

Notacja beznawiasowa a chodzenie po drzewach



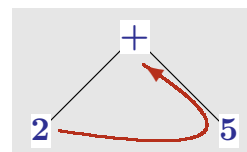
zwykła notacja (*infix*):

$2 + 5$



Notacja Łukasiewicza (*prefix*):

$+ 2 5$

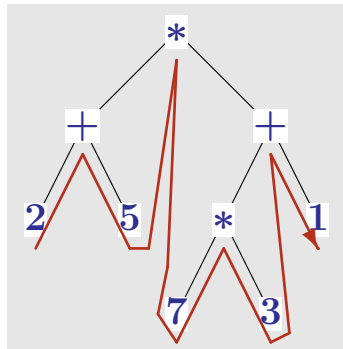


Odwrotna notacja
polska (*postfix*):

$2 5 +$

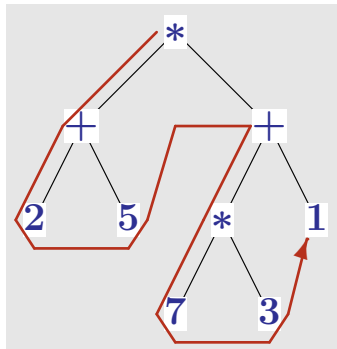
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



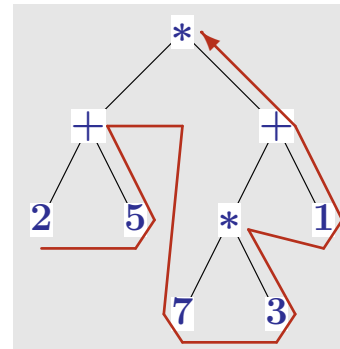
infix — notacja zwykła, **nawiasowa**:

$(2 + 5) * (7 * 3 + 1)$



prefix — notacja Łukasiewicza:

$* + 2 5 + * 7 3 1$

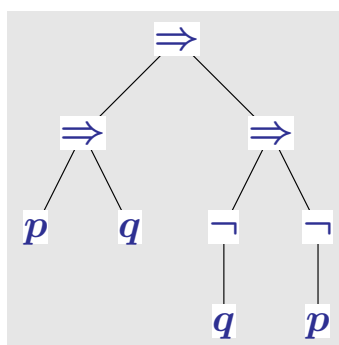


postfix — notacja odwrotna polska:

$2 5 + 7 3 * 1 + *$

Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (wyrażenia logiczne)



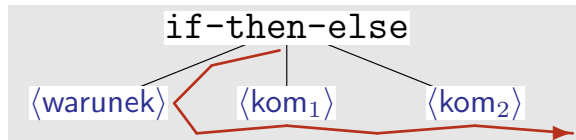
Zwykła notacja (**infix**): $(p \Rightarrow q) \Rightarrow (\neg q \Rightarrow \neg p)$

Notacja Łukasiewicza (**prefix**): $\Rightarrow \Rightarrow p q \Rightarrow \neg q \neg p$

Odwrotna notacja polska (**postfix**): $p q \Rightarrow q \neg p \neg \Rightarrow \Rightarrow$

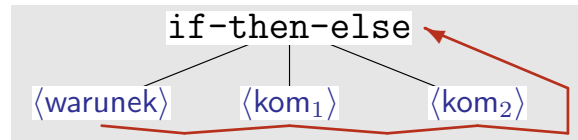
Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (operatory inne niż binarne)



prefix:

if-then-else <warunek> <kom₁> <kom₂>



postfix:

<warunek> <kom₁> <kom₂> if-then-else

infix:

nie da się zapisać nie rozbijając operatora (*distfix*)

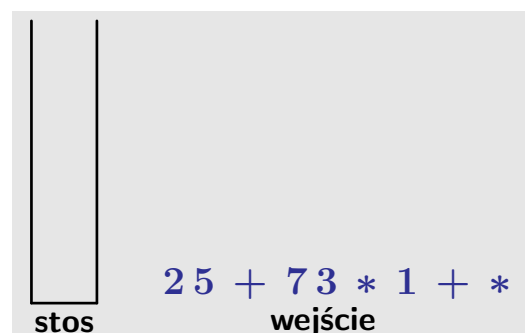
Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.

Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\text{if}(\dots) \dots \text{else} \dots) = 3$.

Algorytm:

```
do {
  jeśli na wejściu jest liczba  $\ell$ ,
    to przełóż ją z wejścia na stos;
  jeśli na wejściu jest operator op,
    to {
      zdejmij ze stosu  $\mathbb{A}(op)$  liczb,
      zastosuj do nich operację op,
      jej wynik włóż na stos;
    }
} while (jest jeszcze coś na wejściu);
```



Działania na językach

— takie jak podstawowe konstrukcje wyrażeń regularnych:

1. Suma:

$$L \mid M \stackrel{\text{def}}{=} \{w \mid w \in L \text{ lub } w \in M\}$$

2. Złączenie:

$$L M \stackrel{\text{def}}{=} \{uv \mid u \in L \text{ i } v \in M\}$$

3. Iteracja (domknięcie Kleenego):

$$L^* \stackrel{\text{def}}{=} \{u_1 u_2 \dots u_n \mid u_1, u_2, \dots, u_n \in L \text{ i } n \geq 0\}$$

Działania na językach

PRZYKŁAD: (suma)

$$L \mid M \stackrel{\text{def}}{=} \{w \mid w \in L \text{ lub } w \in M\}$$

- $\{\text{nie, chce, mi}\} \mid \{\text{się, uczyć}\} = \{\text{nie, chce, mi, się, uczyć}\}$
- $\{\lambda, a, a^2, a^3, \dots\} \mid \{\lambda, b, b^2, b^3, \dots\} = \{\lambda, a, b, a^2, b^2, a^3, b^3, \dots\}$

PRZYKŁAD: (złączenie)

$$L M \stackrel{\text{def}}{=} \{uv \mid u \in L \text{ i } v \in M\}$$

- $\{\text{czerwony, zielony}\} \{\text{dom, kapelusz, rower}\} =$
 $\left\{ \begin{array}{l} \text{czerwonydom, czerwonykapelusz, czerwonyrower,} \\ \text{zielonydom, zielonykapelusz, zielonyrower} \end{array} \right\}$
- $\{\lambda, a, a^2, a^3, \dots\} \{\lambda, b, b^2, b^3, \dots\} =$
 $\{\lambda, a, b, aa, ab, bb, aaa, aab, abb, bbb, \dots\}$

Działania na językach

Własności działań:

- Łączność:

$$K \mid (L \mid M) = (K \mid L) \mid M \qquad K(LM) = (KL)M$$

- Przemienność:

$$L \mid M = M \mid L$$

Uwaga: złączenie nie jest przemienne!

- Elementy neutralne:

$$M \mid \emptyset = M \qquad \Lambda M = M \Lambda = M$$

$\emptyset \stackrel{\text{def}}{=} \{\}$ — język pusty

$\Lambda \stackrel{\text{def}}{=} \{\lambda\}$ — język zawierający tylko słowo puste

- Rozdzielność:

$$K(L \mid M) = KL \mid KM \qquad (L \mid M)K = LK \mid MK$$

Działania na językach

Podobieństwa do arytmetyki:

- **suma** języków jest jak **dodawanie** liczb,
język $\emptyset$ jest jak liczba 0 ;
- **złączenie** języków jest jak **mnożenie** liczb,
język Λ jest jak liczba 1 .

Różnice:

- złączenie języków *nie jest* przemienne;
- *nie ma* działań odwrotnych do sumy ani złączenia (czyli odpowiedników odejmowania i dzielenia).

Działania na językach

Podobieństwa do arytmetyki:

To, co robimy w arytmetyce z użyciem odejmowania i dzielenia, czasem daje się wykonać bez nich (choć w bardziej skomplikowany sposób).

PRZYKŁAD:

Rozwiązaniem równania $x = a \cdot x + b$ jest **suma ciągu geometrycznego**

$$x = b \cdot \frac{1}{1-a} = b \cdot \lim_{n \rightarrow \infty} \frac{1-a^n}{1-a} = b \cdot \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} a^i = b \cdot \sum_{i=0}^{\infty} a^i$$

dzielenie (na $1-a$), *odejmowanie* (z 1), *jeśli $a < 1$*

Zamiast odejmowania i dzielenia wystarczy mnożenie i *nieskończone* dodawanie **(w tym przypadku)**.

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \{u_1 u_2 \dots u_n \mid u_1, u_2, \dots, u_n \in L \text{ i } n \geq 0\}$$

$$L^* = \Lambda \mid L \mid L^2 \mid L^3 \mid \dots = \{\lambda\} \mid L \mid LL \mid LLL \mid \dots$$

Przykładowe własności:

- $(L \mid M)^* = (L^* M^*)^* = (L^* M)^* L^*$
- $(LM)^* = \Lambda \mid L(ML)^* M$

Iteracja do pewnego stopnia zastępuje „odejmowanie” i „dzielenie” na raz:

$$\frac{1}{1-M} \longleftrightarrow M^*$$

$$\frac{1}{1-M} = 1 + \frac{1}{1-M} \cdot M \longleftrightarrow M^* = \Lambda \mid M^* M$$

$$\frac{L}{1-M} = L + \frac{L}{1-M} \cdot M \longleftrightarrow L M^* = L \mid L M^* M$$

$$\begin{array}{l} \text{równanie: } X = L + X \cdot M \\ \text{rozwiązanie: } X = L \cdot \frac{1}{1-M} \end{array}$$

$\longleftrightarrow$

$$\begin{array}{l} \text{równanie: } X = L \mid X M \\ \text{rozwiązanie: } X = L M^* \end{array}$$

Eliminacja lewostronnej rekursji z gramatyki

Iteracja:

$$L^* \stackrel{\text{def}}{=} \{u_1 u_2 \dots u_n \mid u_1, u_2, \dots, u_n \in L \text{ i } n \geq 0\}$$

$$L^* = \Lambda \mid L \mid L^2 \mid L^3 \mid \dots = \{\lambda\} \mid L \mid LL \mid LLL \mid \dots$$

TWIERDZENIE: (rozwiązywanie równania językowego)

Rozwiązaniem równania $X = L \mid XM$ jest $X = LM^*$.

lewostronna rekursja

Powyżej:

- X — niewiadoma (szukany język)
- L i M — wyrażenia językowe **nie zaczynające się od X** (mogą zawierać X wewnątrz)

Przekształcenia gramatyk

Interpretujemy

- *nieterminale* jako języki złożone z tych słów, które dają się z nich wyprowadzić;
- *terminale* jako języki 1-literowe;
- $\text{znak} ::=$ jako równość języków.

Wtedy gramatyka staje się *układem równań* na językach, przy czym nieterminale są jego *niewiadomymi*.

PRZYKŁAD:

$$\langle \text{Id} \rangle ::= a \mid b \mid \langle \text{Id} \rangle a \mid \langle \text{Id} \rangle b \mid \langle \text{Id} \rangle 0 \mid \langle \text{Id} \rangle 1$$



— łączność i rozdzielność

$$\langle \text{Id} \rangle ::= \{a \mid b\} \mid \langle \text{Id} \rangle \{a \mid b \mid 0 \mid 1\}$$



— tw. o rozw. równania językowego —

$$\langle \text{Id} \rangle ::= \{a \mid b\} \{a \mid b \mid 0 \mid 1\}^*$$

$$\langle X \rangle = L \mid \langle X \rangle M$$



$$\langle X \rangle = L M^*$$