Stefan Sokołowski

JĘZYKI FORMALNE I METODY KOMPILACJI

wykład 5

Inst. Informatyki Stosowanej, ANS
Elbląg, 2024/2025

Notacja beznawiasowa Łukasiewicza

Notacja Łukasiewicza (w świecie nazywana *Polish*):
Wyrażenia (arytmetyczne, logiczne, itp.) można zapisywać **bez nawiasów** w postaci

$$\texttt{operator}\ \texttt{arg}_1\ \texttt{arg}_2\ \ldots\ \texttt{arg}_n$$

Jan Łukasiewicz
1878–1956

Notacja beznawiasowa Łukasiewicza

Notacja Łukasiewicza (w świecie nazywana *Polish*):
Wyrażenia (arytmetyczne, logiczne, itp.) można zapisywać **bez nawiasów** w postaci

$$\texttt{operator}\ \texttt{arg}_1\ \texttt{arg}_2\ \ldots\ \texttt{arg}_n$$

Odwrotna notacja Łukasiewicza (*Reverse Polish*):

$$\texttt{arg}_1\ \texttt{arg}_2\ \ldots\ \texttt{arg}_n\ \texttt{operator}$$

stosowana w informatyce.

Jan Łukasiewicz
1878–1956

Jak to czytać?

Jak to czytać?

(na czerwono pod operatorem — liczba argumentów czyli „arność")

Łukasiewicza	odwrotna	czyli	klasycznie
$\underset{2}{+}\ 2\ 5$			

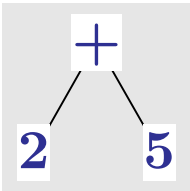
Jak to czytać?

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

Łukasiewicza	odwrotna	czyli	klasycznie
$\underset{2}{+}\ 2\ 5$	$2\ 5\ \underset{2}{+}$		

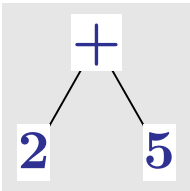
Jak to czytać?

(na czerwono pod operatorem — liczba argumentów czyli „arność")

Łukasiewicza	odwrotna	czyli	klasycznie
$\underset{2}{+}\ 2\ 5$	$2\ 5\ \underset{2}{+}$		

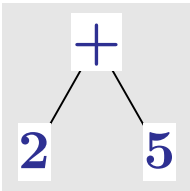
Jak to czytać?

(na czerwono pod operatorem — liczba argumentów czyli „arność")

Łukasiewicza	odwrotna	czyli	klasycznie
$\underset{2}{+}\ 2\ 5$	$2\ 5\ \underset{2}{+}$		$2\ +\ 5$

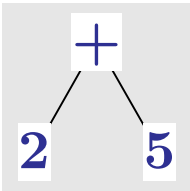
Jak to czytać?

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

Łukasiewicza	odwrotna	czyli	klasycznie
$\underset{2}{+}\ 2\ 5$	$2\ 5\ \underset{2}{+}$		$2\ +\ 5$
$*\ \underset{2}{+}\ 2\ 5\ \underset{2}{-}\ 3\ 1$			

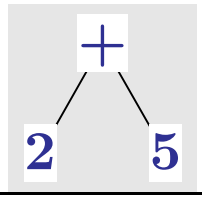
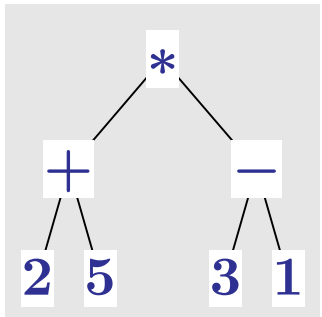
Jak to czytać?

(na czerwono pod operatorem — liczba argumentów czyli „arność")

Łukasiewicza	odwrotna	czyli	klasycznie
$\underset{2}{+}\ 2\ 5$	$2\ 5\ \underset{2}{+}$		$2\ +\ 5$
$\underset{2}{*}\ \underset{2}{+}\ 2\ 5\ \underset{2}{-}\ 3\ 1$	$2\ 5\ \underset{2}{+}\ 3\ 1\ \underset{2}{-}\ \underset{2}{*}$		

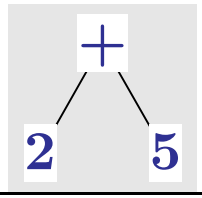
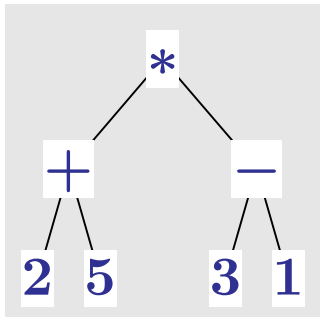
Jak to czytać?

(na czerwono pod operatorem — liczba argumentów czyli „arność")

Łukasiewicza	odwrotna	czyli	klasycznie
$\underset{2}{+}\ 2\ 5$	$2\ 5\ \underset{2}{+}$		$2\ +\ 5$
$\underset{2}{*}\ \underset{2}{+}\ 2\ 5\ \underset{2}{-}\ 3\ 1$	$2\ 5\ \underset{2}{+}\ 3\ 1\ \underset{2}{-}\ \underset{2}{*}$		

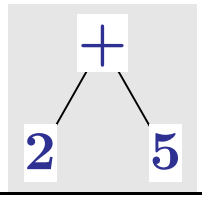
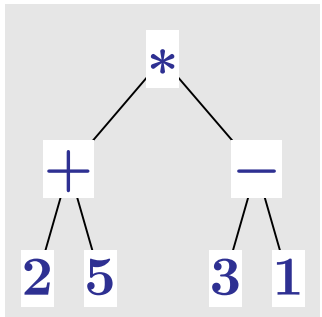
Jak to czytać?

(na czerwono pod operatorem — liczba argumentów czyli „arność")

Łukasiewicza	odwrotna	czyli	klasycznie
$\underset{2}{+}\ 2\ 5$	$2\ 5\ \underset{2}{+}$		$2\ +\ 5$
$\underset{2}{*}\ \underset{2}{+}\ 2\ 5\ \underset{2}{-}\ 3\ 1$	$2\ 5\ \underset{2}{+}\ 3\ 1\ \underset{2}{-}\ \underset{2}{*}$		$(\ 2\ +\ 5\ )\ *\ (\ 3\ -\ 1\ )$

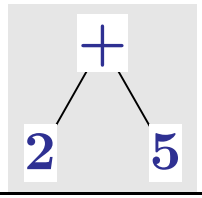
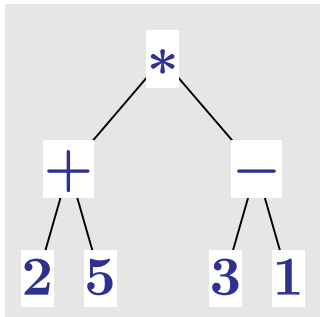
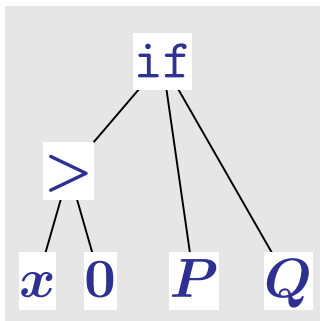
Jak to czytać?

(na czerwono pod operatorem — liczba argumentów czyli „arność")

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

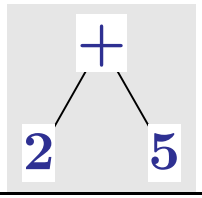
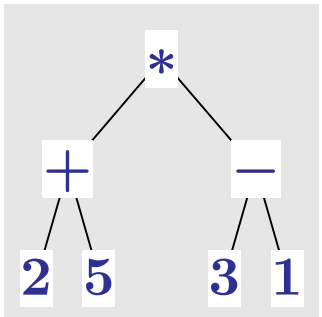
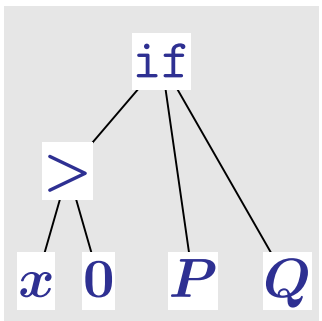
Jak to czytać?

(na czerwono pod operatorem — liczba argumentów czyli „arność")

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

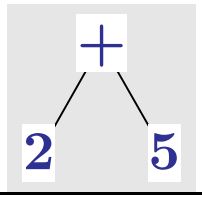
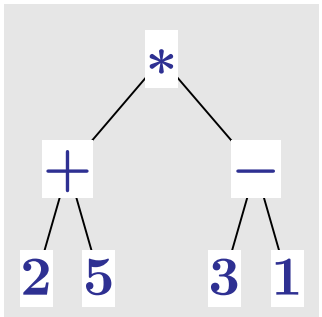
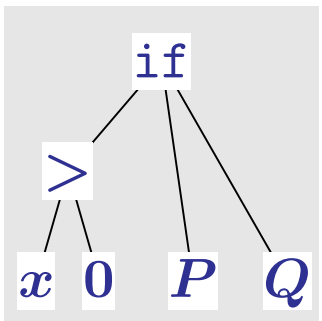
Jak to czytać?

(na czerwono pod operatorem — liczba argumentów czyli „arność")

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

Jak to czytać?

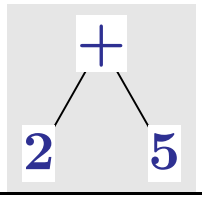
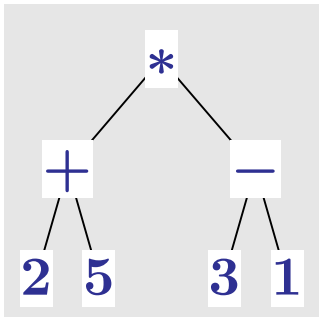
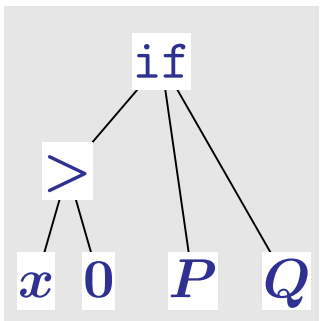
(na czerwono pod operatorem — liczba argumentów czyli „arność")

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

Wyrażenia — drzewa (nie całkiem drz. wywodu).

Jak to czytać?

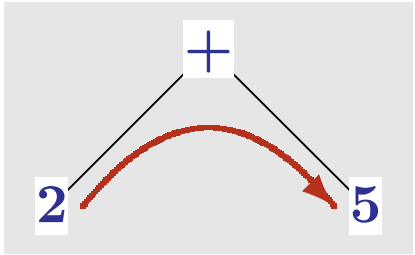
(na czerwono pod operatorem — liczba argumentów czyli „arność")

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

Wyrażenia — drzewa (nie całkiem drz. wywodu).
Różne **notacje** — sposoby chodzenia po drzewach.

Notacja beznawiasowa a chodzenie po drzewach

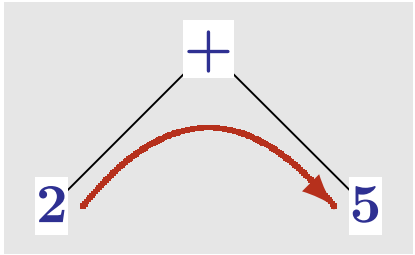
Notacja beznawiasowa a chodzenie po drzewach



zwykła notacja *(infix)*:

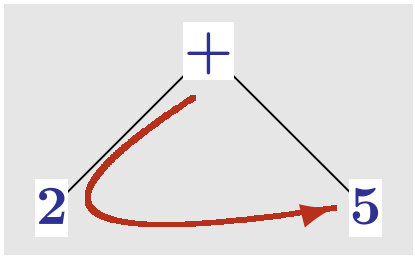
$2 + 5$

Notacja beznawiasowa a chodzenie po drzewach



zwykła notacja *(infix)*:

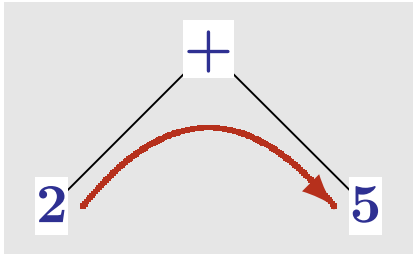
$2 + 5$



Notacja Łukasiewicza *(prefix)*:

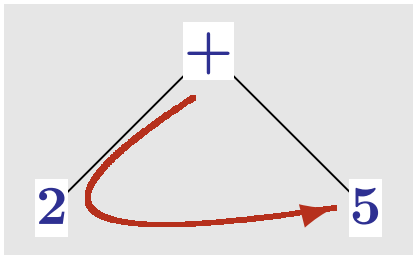
$+\ 2\ 5$

Notacja beznawiasowa a chodzenie po drzewach



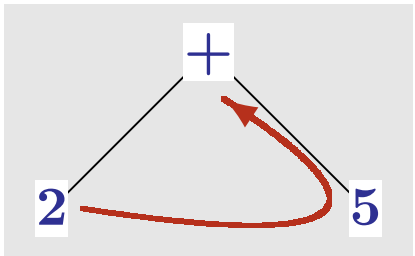
zwykła notacja *(infix)*:

$2 + 5$



Notacja Łukasiewicza *(prefix)*:

$+\ 2\ 5$

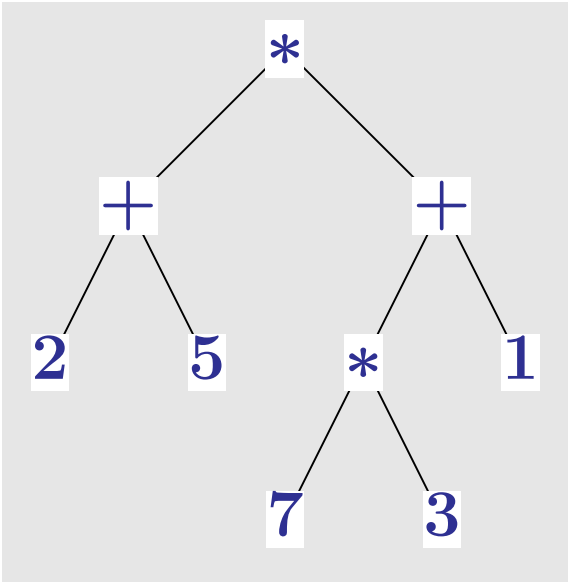


Odwrotna notacja polska *(postfix)*:

$2\ 5\ +$

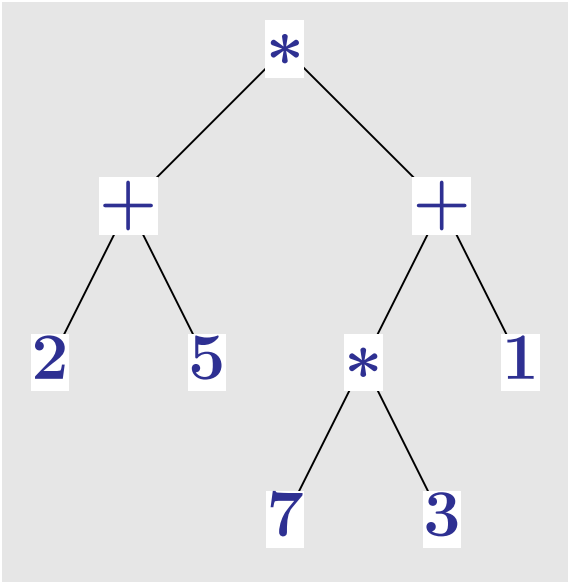
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



Notacja beznawiasowa a chodzenie po drzewach

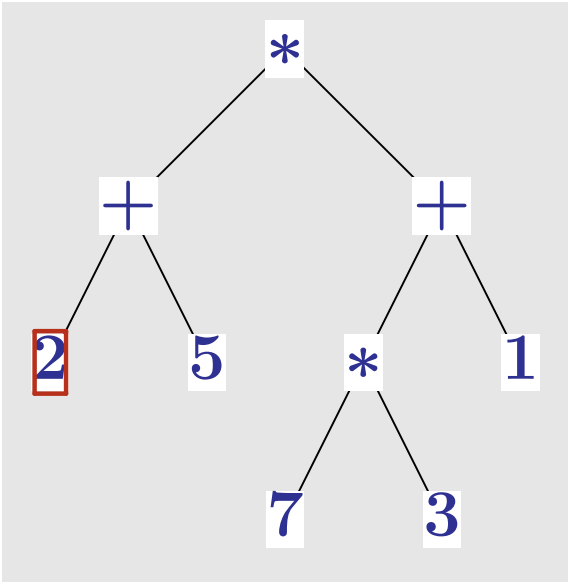
PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa**

Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)

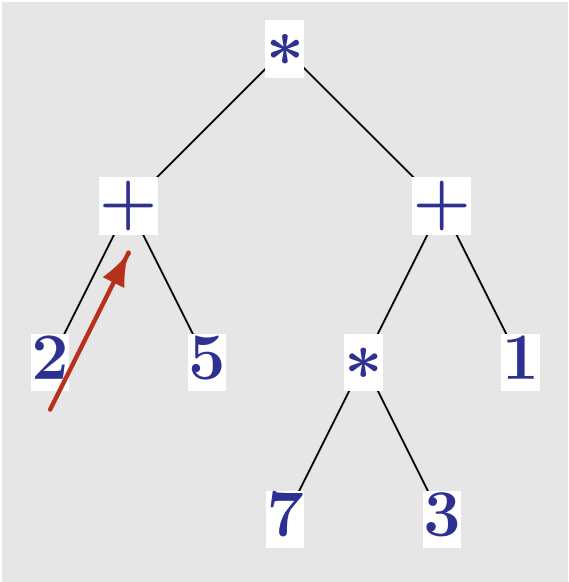


infix **— notacja zwykła, nawiasowa:**

2

Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)

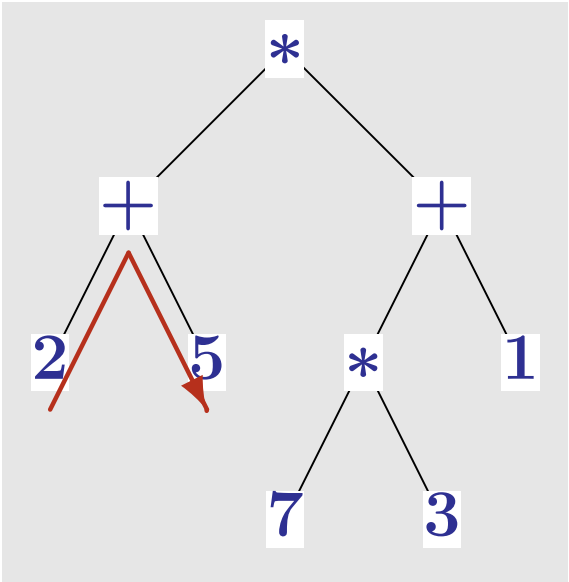


infix **— notacja zwykła, nawiasowa:**

$2 +$

Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)

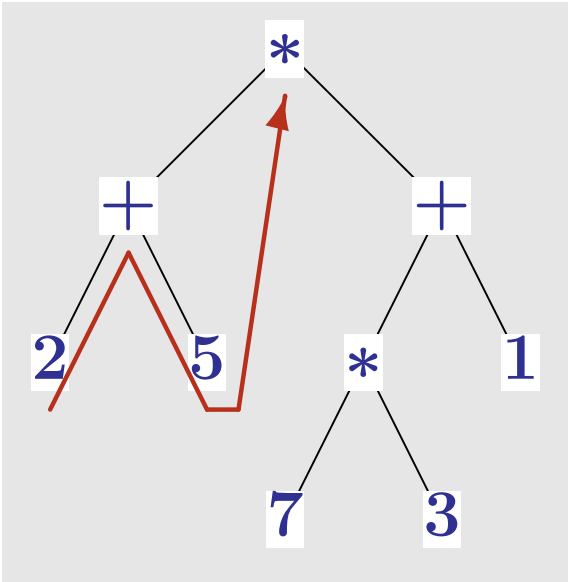


infix **— notacja zwykła, nawiasowa:**

$$2 + 5$$

Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)

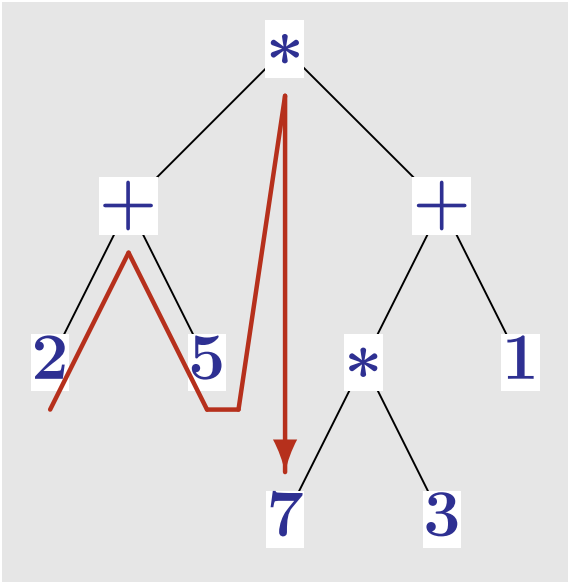


infix **— notacja zwykła, nawiasowa:**

$(2 + 5) *$

Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)

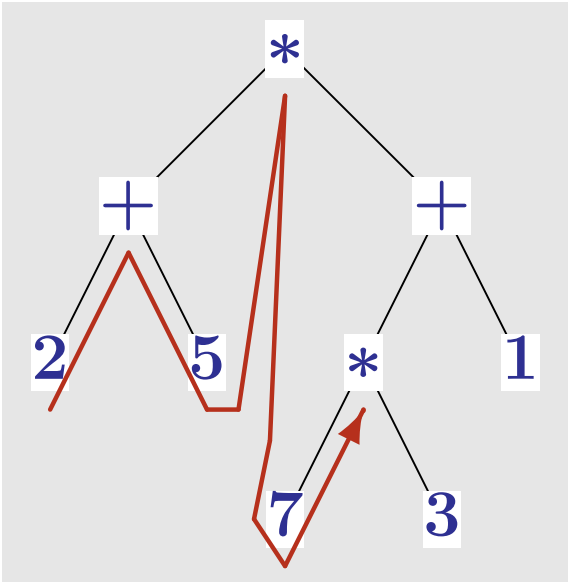


infix **— notacja zwykła, nawiasowa:**

$(2 + 5) * (7 \qquad)$

Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)

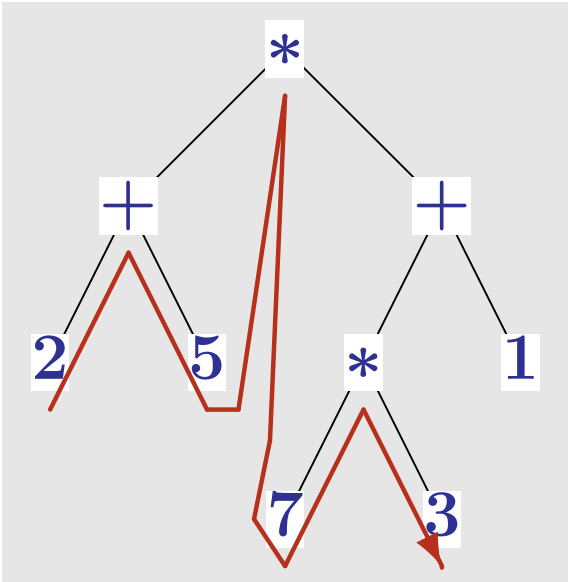


infix **— notacja zwykła, nawiasowa:**

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

Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)

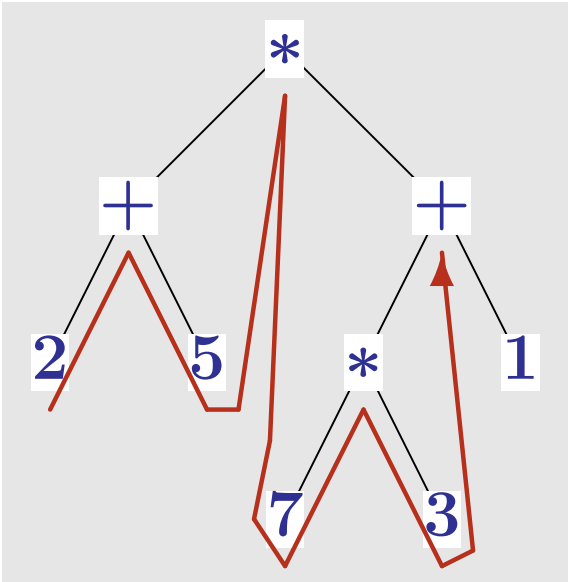


infix **— notacja zwykła, nawiasowa:**

$$(2+5)*(7*3\qquad)$$

Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)

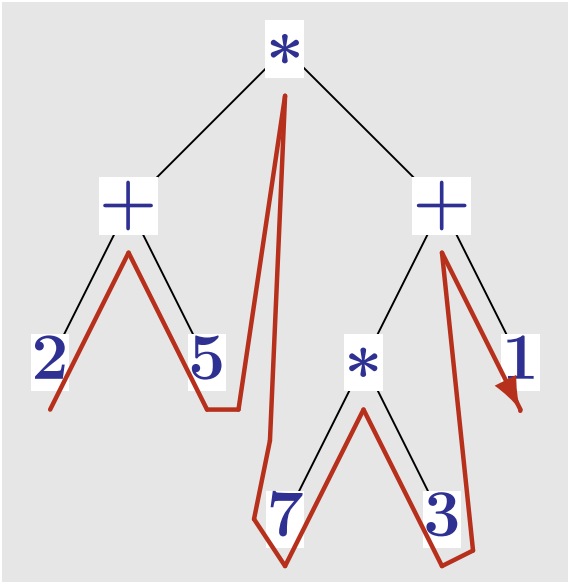


infix **— notacja zwykła, nawiasowa:**

$$(2+5)*(7*3+\quad)$$

Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)

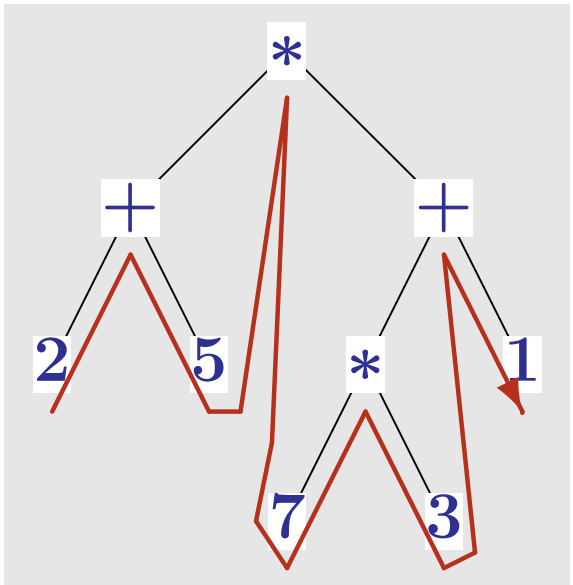


infix **— notacja zwykła, nawiasowa:**

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

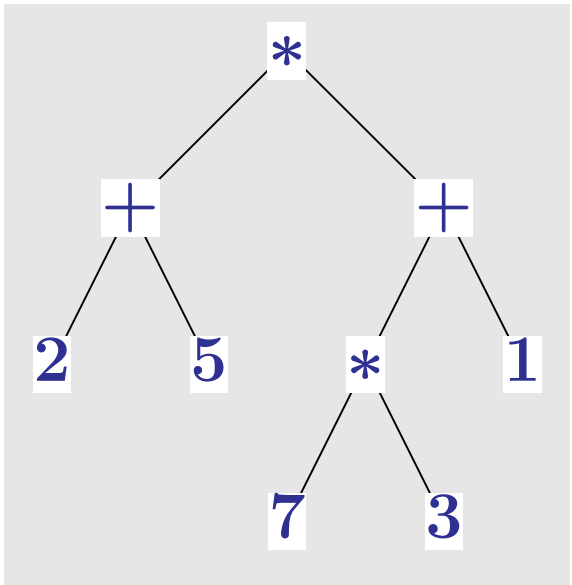
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

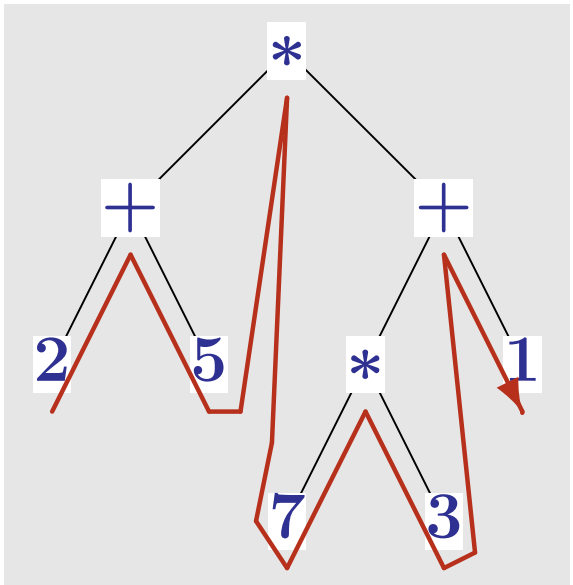
$$(2 + 5) * (7 * 3 + 1)$$



prefix **— notacja Łukasiewicza**

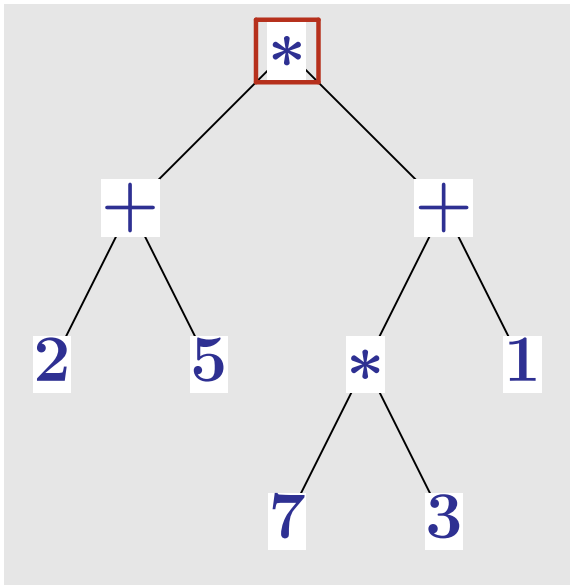
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

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

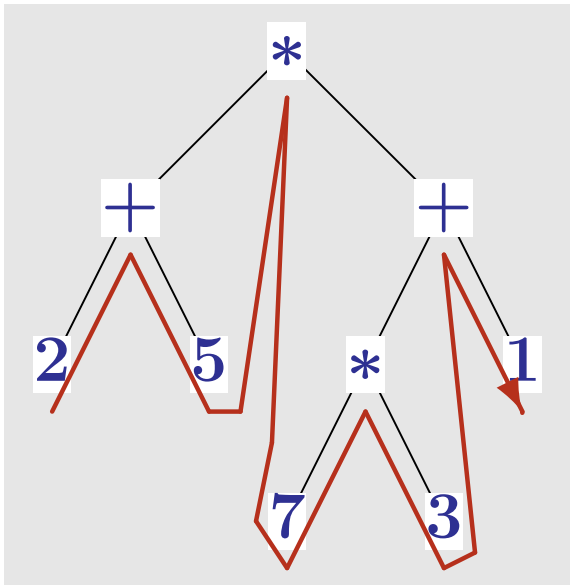


prefix **— notacja Łukasiewicza:**

$$*$$

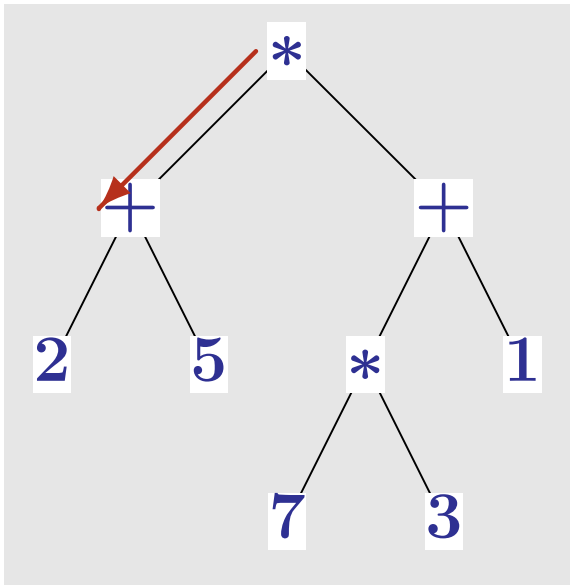
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

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

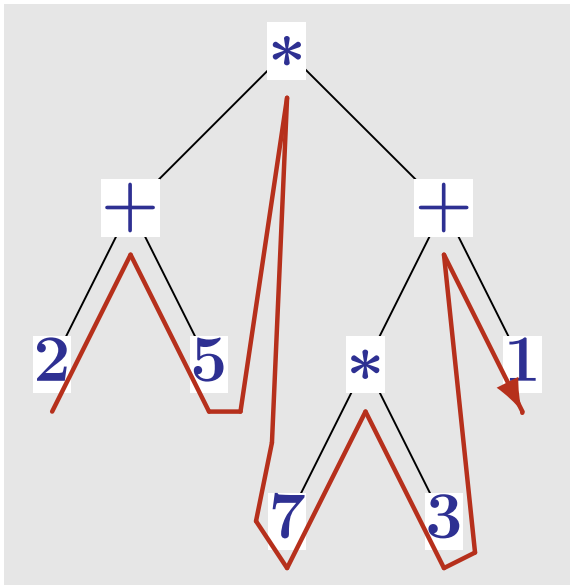


prefix **— notacja Łukasiewicza:**

$$*\ +$$

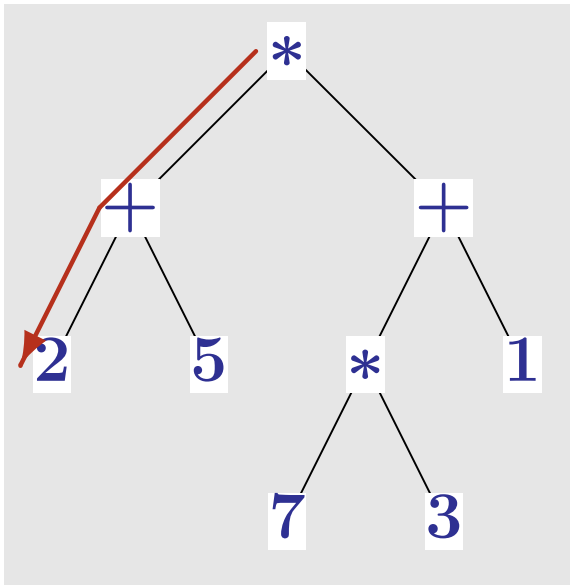
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

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

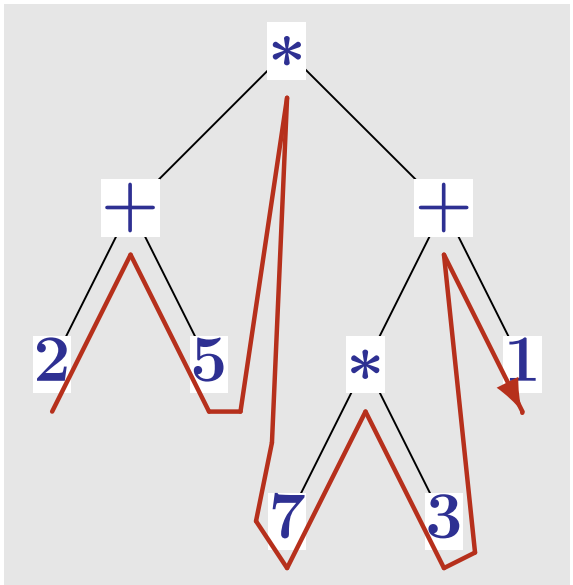


prefix **— notacja Łukasiewicza:**

$$* \ + \ 2$$

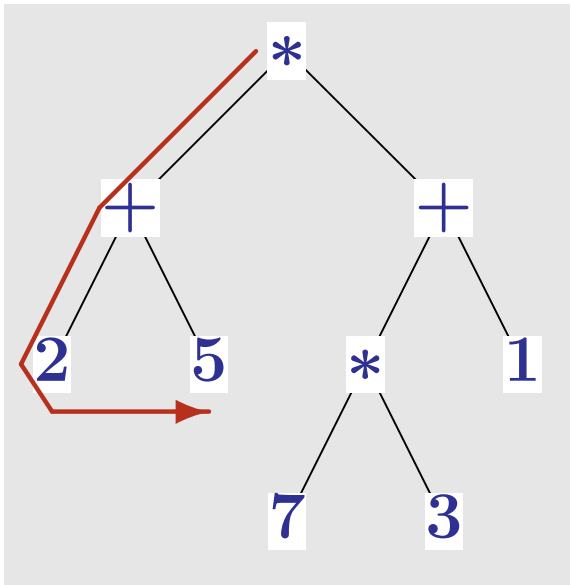
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

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

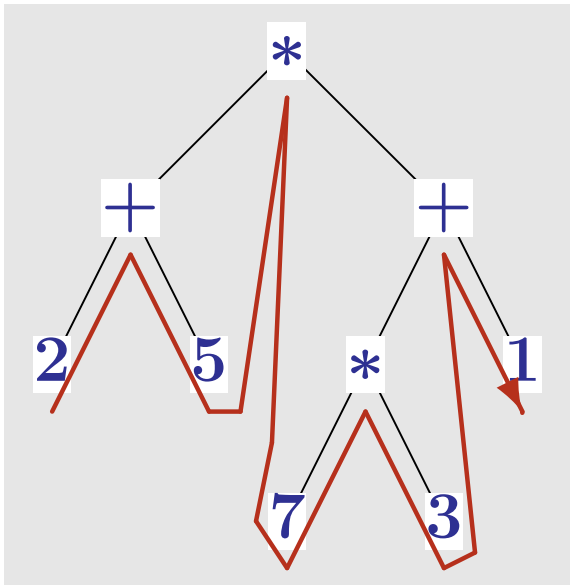


prefix **— notacja Łukasiewicza:**

$$* \; + \; 2 \; 5$$

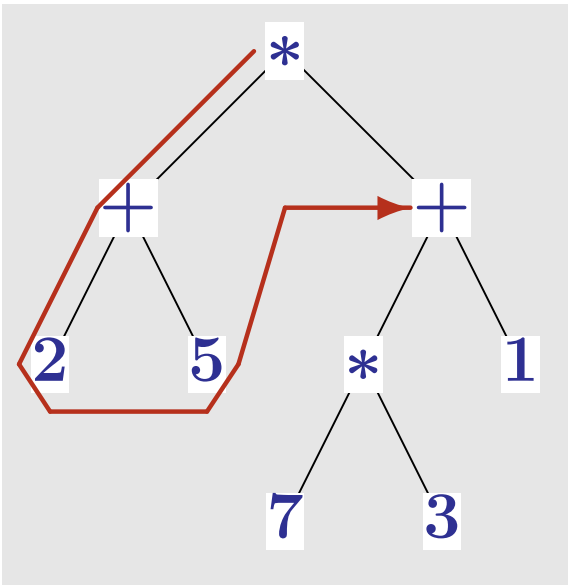
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

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

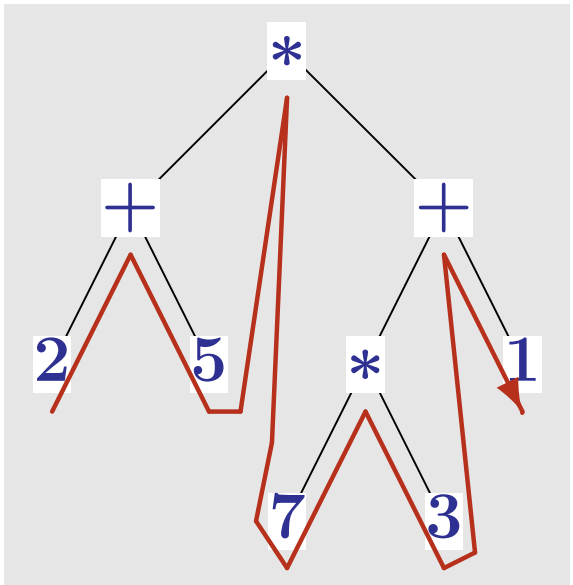


prefix **— notacja Łukasiewicza:**

$$* \; + \; 2 \; 5 \; + \; * \; 7 \; 3 \; 1$$

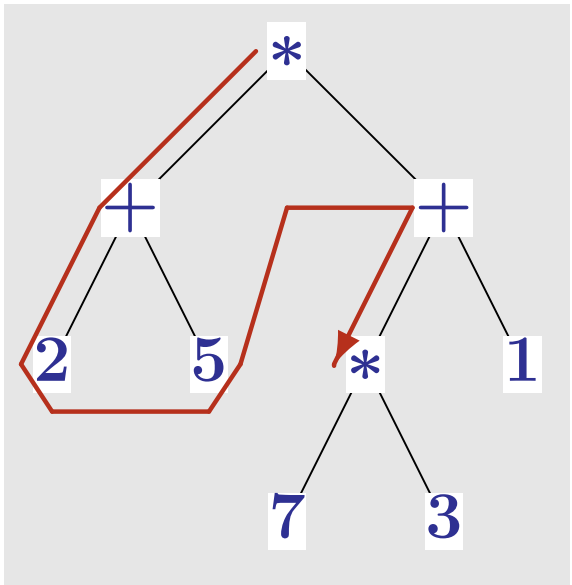
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

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

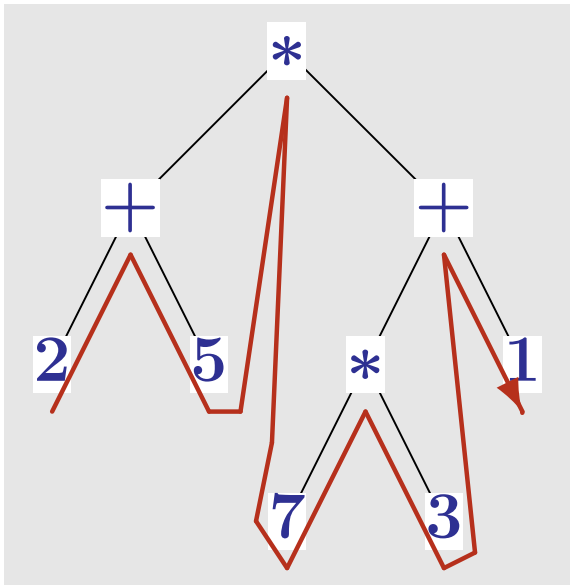


prefix **— notacja Łukasiewicza:**

$$* \; + \; 2 \; 5 \; + \; *$$

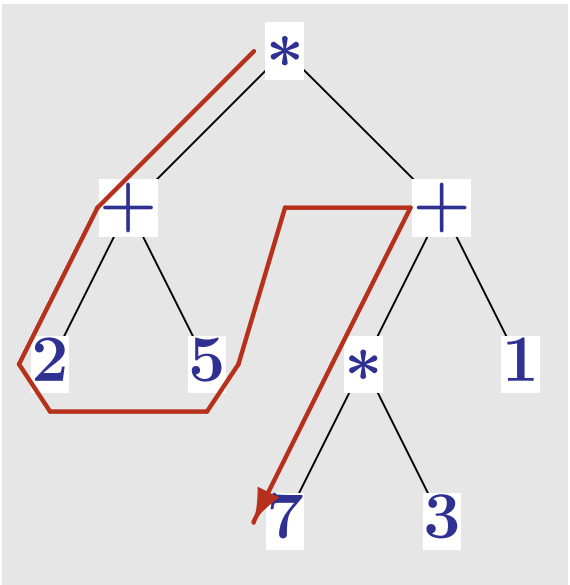
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

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

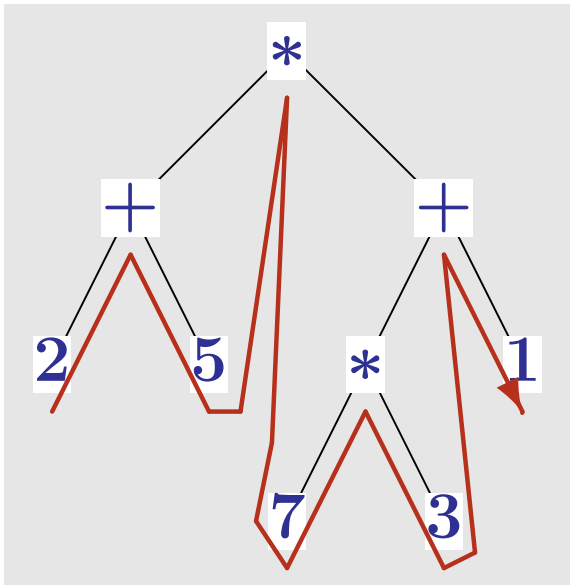


prefix **— notacja Łukasiewicza:**

$$* \; + \; 2 \; 5 \; + \; * \; 7$$

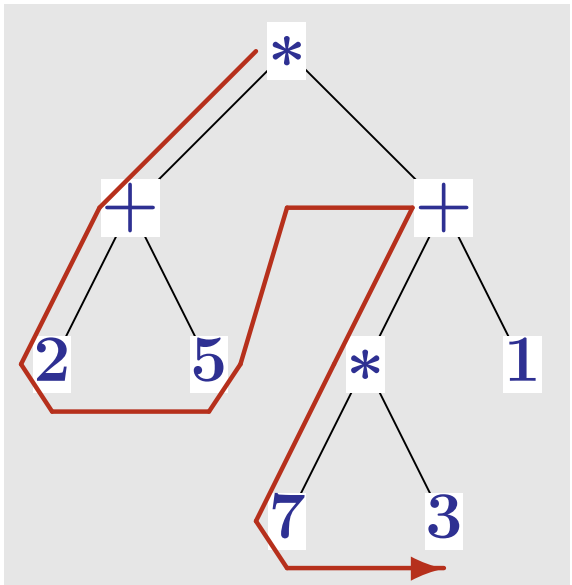
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

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

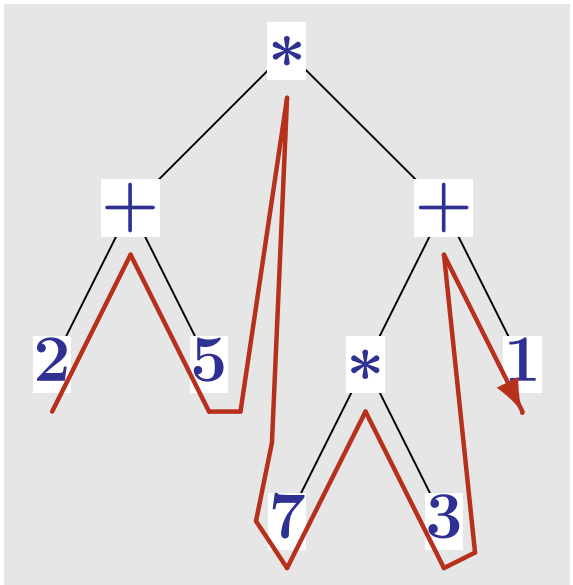


prefix **— notacja Łukasiewicza:**

$$* \; + \; 2 \; 5 \; + \; * \; 7 \; 3$$

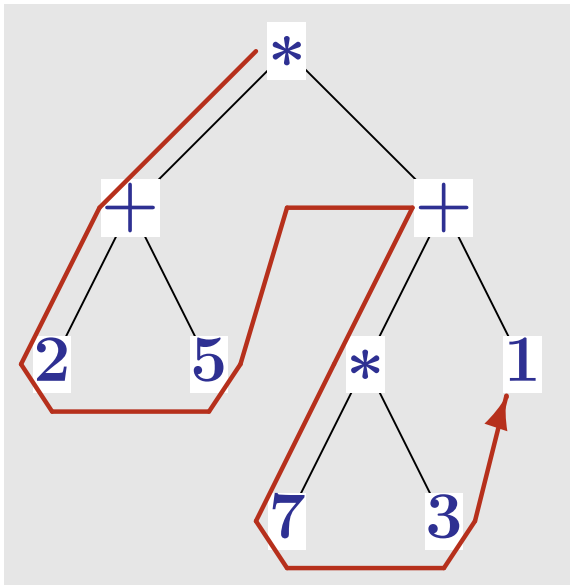
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



infix **— notacja zwykła, nawiasowa:**

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

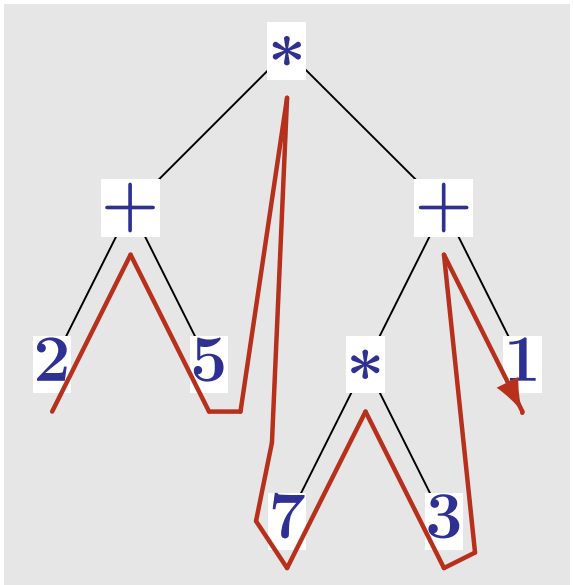


prefix **— notacja Łukasiewicza:**

$$*\ +\ 2\ 5\ +\ *\ 7\ 3\ 1$$

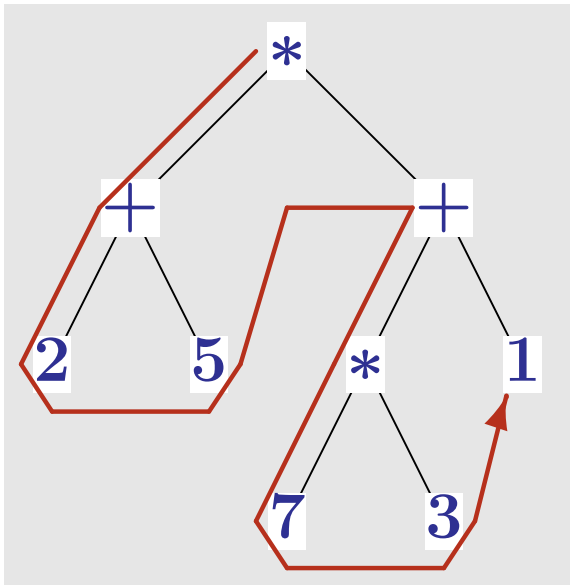
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



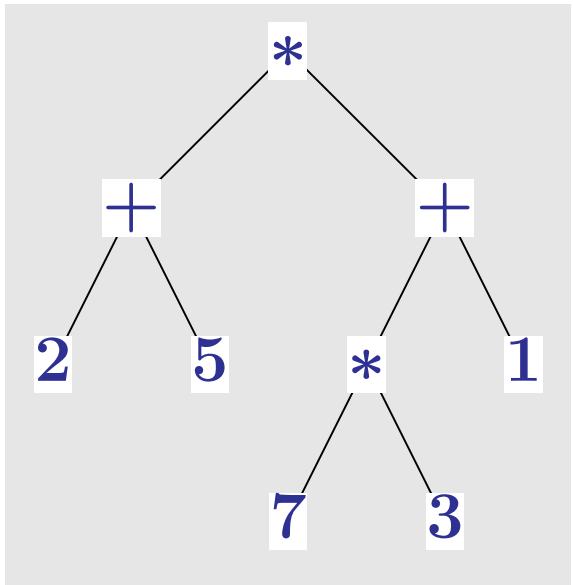
infix **— notacja zwykła, nawiasowa:**

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



prefix **— notacja Łukasiewicza:**

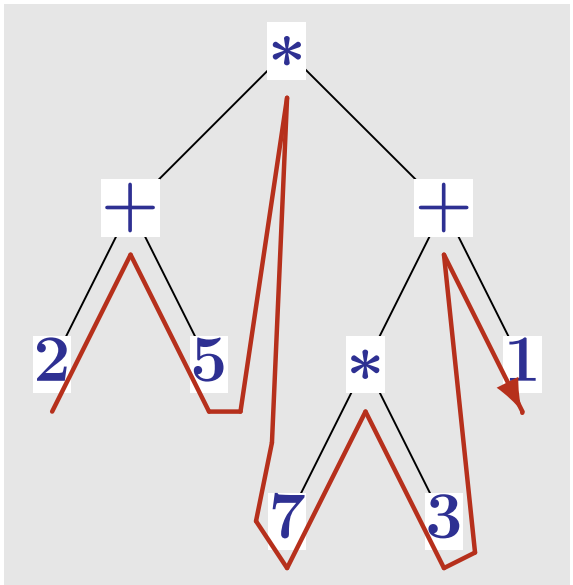
$$*\ +\ 2\ 5\ +\ *\ 7\ 3\ 1$$



postfix **— notacja odwrotna polska**

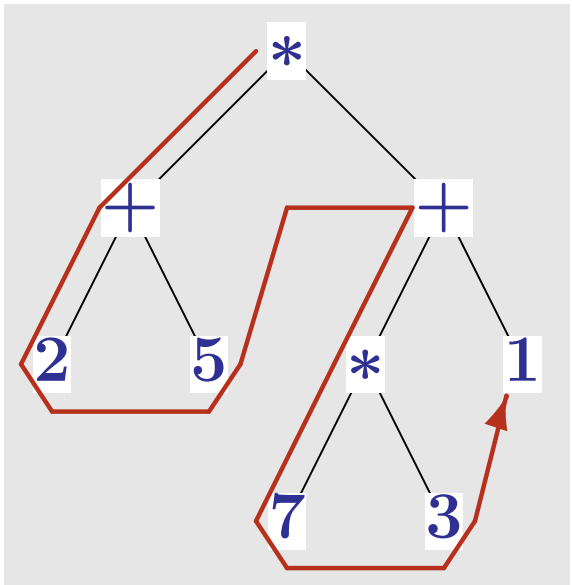
Notacja beznawiasowa a chodzenie po drzewach

PRZYKŁAD: (bardziej złożony)



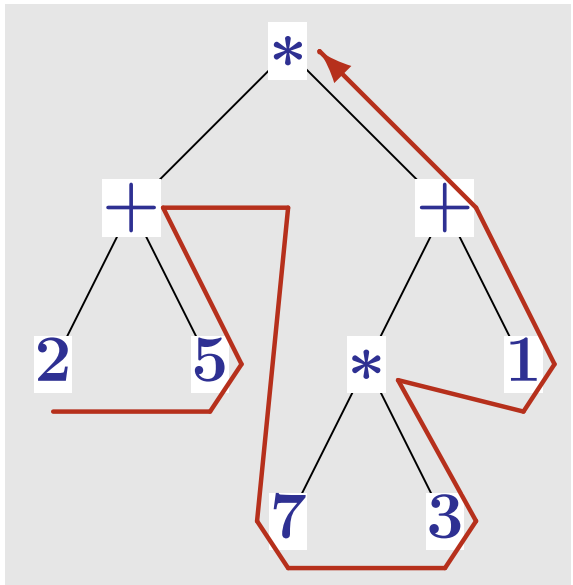
infix **— notacja zwykła, nawiasowa:**

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



prefix **— notacja Łukasiewicza:**

$$*\ +\ 2\ 5\ +\ *\ 7\ 3\ 1$$

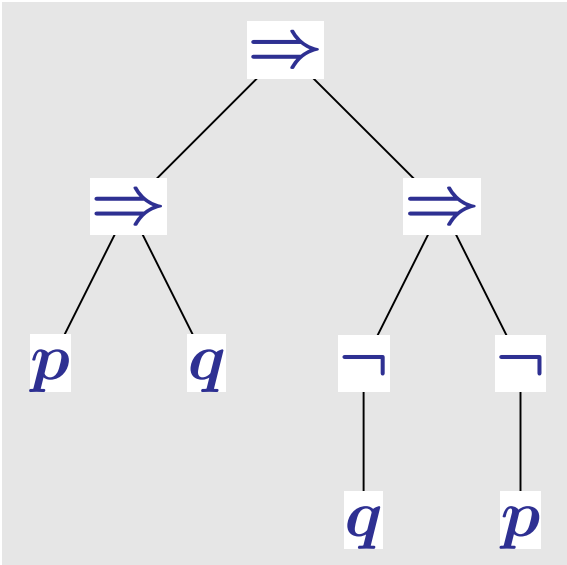


postfix **— notacja odwrotna polska:**

$$2\ 5\ +\ 7\ 3\ *\ 1\ +\ *$$

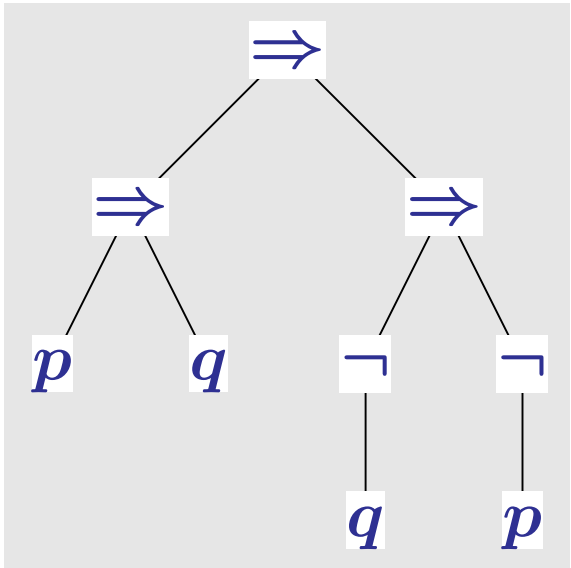
Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (wyrażenia logiczne)



Notacja nawiasowa i beznawiasowa

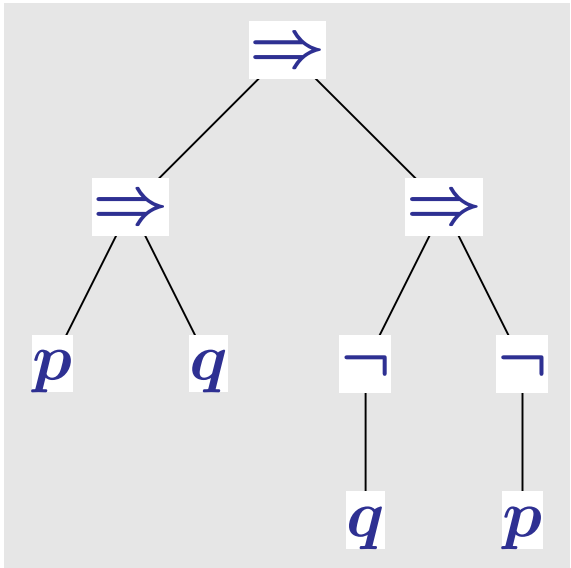
PRZYKŁAD: (wyrażenia logiczne)



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

Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (wyrażenia logiczne)

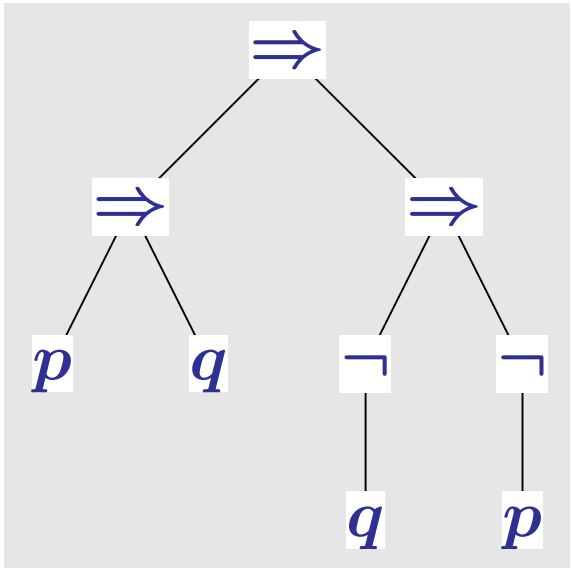


Zwykła notacja *(infix)*: $(p \Rightarrow q) \Rightarrow (\neg q \Rightarrow \neg p)$

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

Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (wyrażenia logiczne)



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

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

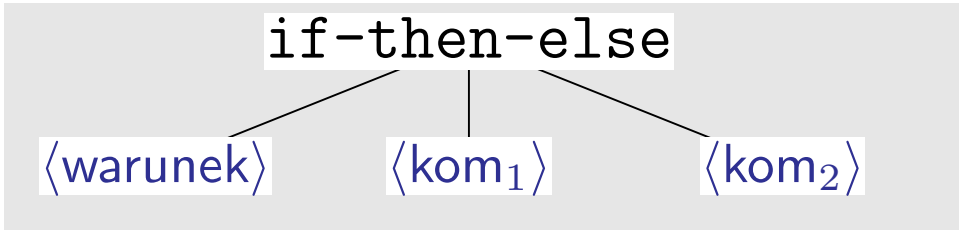
Odwrotna notacja polska *(postfix)*: $p \; q \Rightarrow \;\; q \; \neg \; p \; \neg \;\; \Rightarrow \Rightarrow$

Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (operatory inne niż binarne)

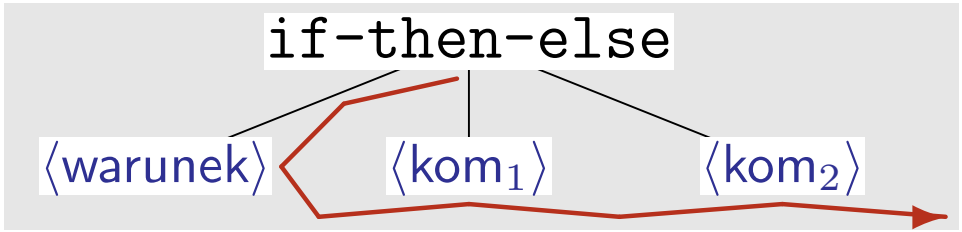
Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (operatory inne niż binarne)



Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (operatory inne niż binarne)

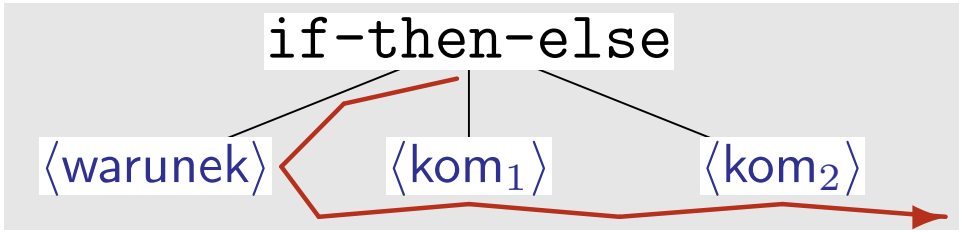


***prefix*:**

`if-then-else` ⟨warunek⟩ ⟨kom$_1$⟩ ⟨kom$_2$⟩

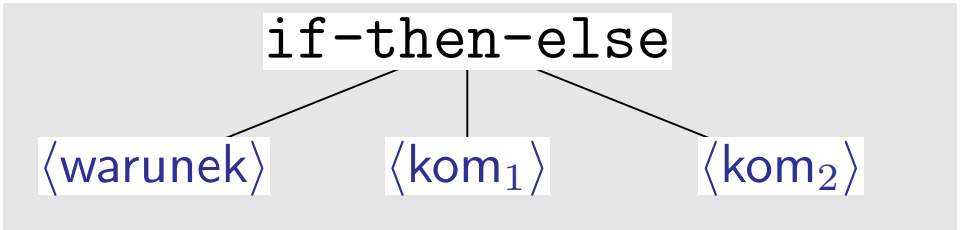
Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (operatory inne niż binarne)



***prefix*:**

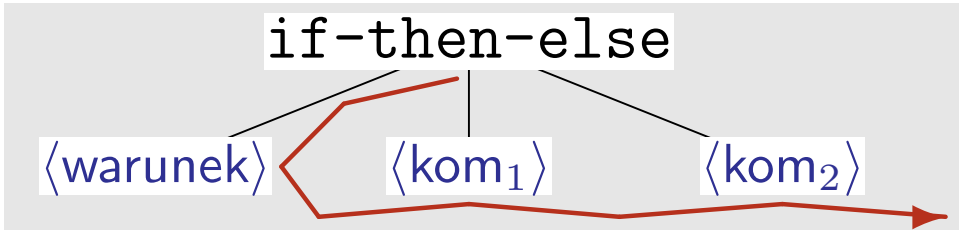
`if-then-else` ⟨warunek⟩ ⟨kom$_1$⟩ ⟨kom$_2$⟩



postfix

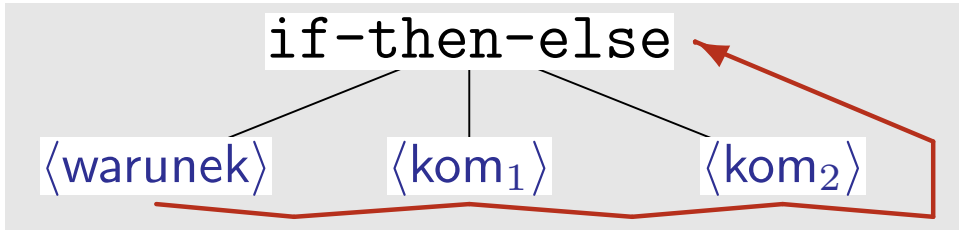
Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (operatory inne niż binarne)



***prefix*:**

`if-then-else` ⟨warunek⟩ ⟨kom$_1$⟩ ⟨kom$_2$⟩

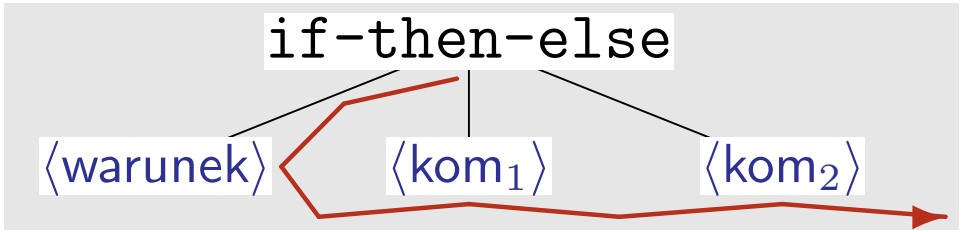


***postfix*:**

⟨warunek⟩ ⟨kom$_1$⟩ ⟨kom$_2$⟩ `if-then-else`

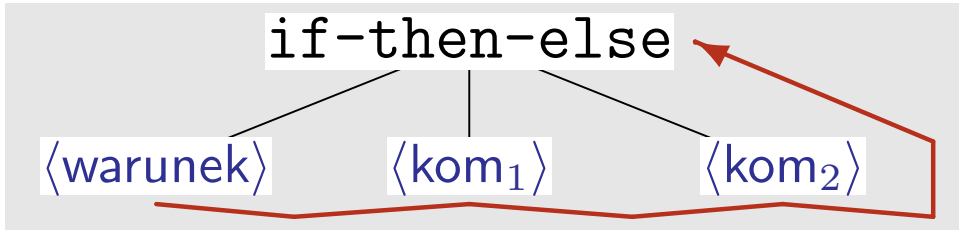
Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (operatory inne niż binarne)



***prefix*:**

`if-then-else` ⟨warunek⟩ ⟨kom$_1$⟩ ⟨kom$_2$⟩



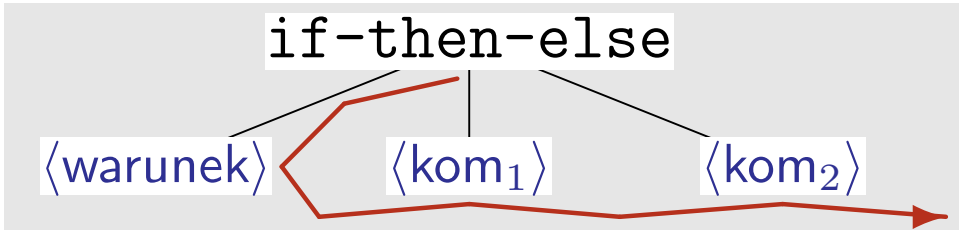
***postfix*:**

⟨warunek⟩ ⟨kom$_1$⟩ ⟨kom$_2$⟩ `if-then-else`

infix

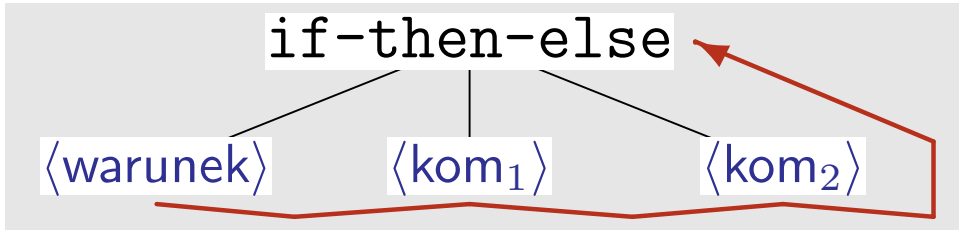
Notacja nawiasowa i beznawiasowa

PRZYKŁAD: (operatory inne niż binarne)



***prefix*:**

`if-then-else` ⟨warunek⟩ ⟨kom$_1$⟩ ⟨kom$_2$⟩



***postfix*:**

⟨warunek⟩ ⟨kom$_1$⟩ ⟨kom$_2$⟩ `if-then-else`

***infix*:**

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

Obliczanie wyrażeń w odwrotnej notacji polskiej

Obliczanie wyrażeń w odwrotnej notacji polskiej

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

Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

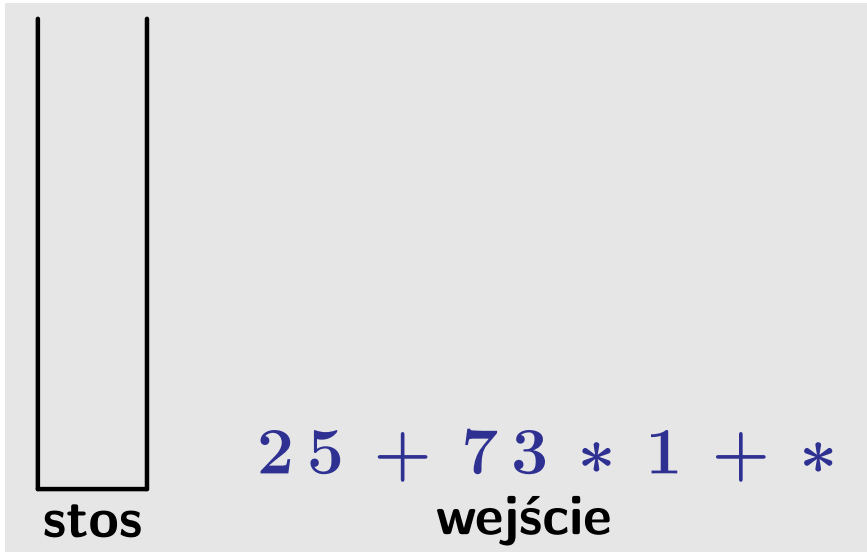
do {
  jeśli na wejściu jest liczba ℓ,
    to przełóż ją z wejścia na stos;
  jeśli na wejściu jest operator *op*,
    to {
      zdejmij ze stosu $\mathbb{A}(op)$ liczb,
      zastosuj do nich operację *op*,
      jej wynik włóż na stos;
    }
} **while** (jest jeszcze coś na wejściu);

Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

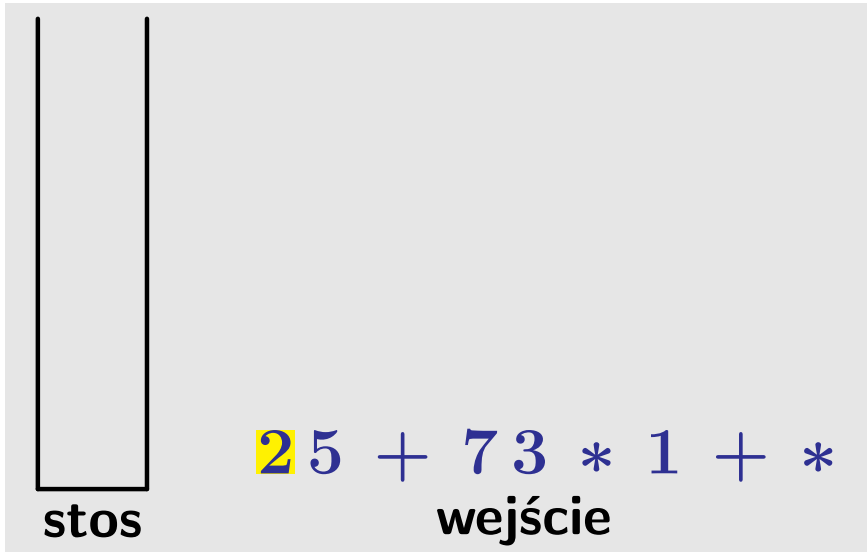


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

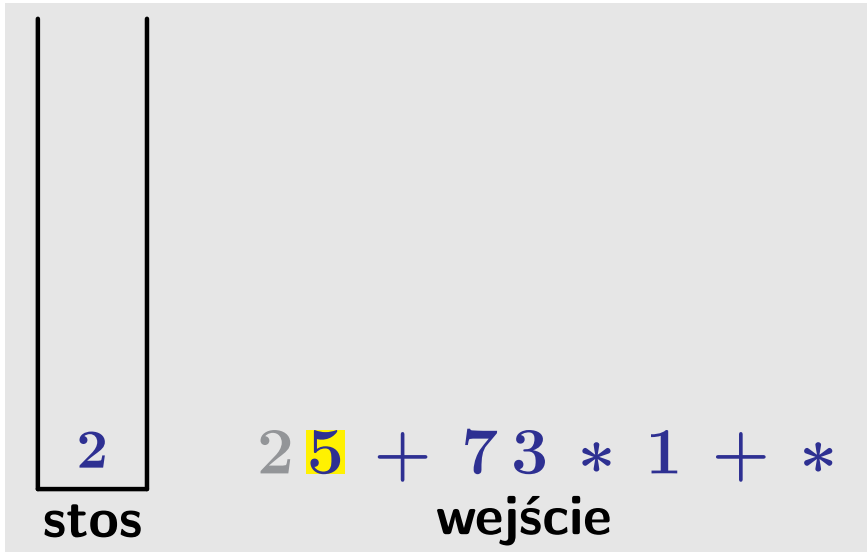


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

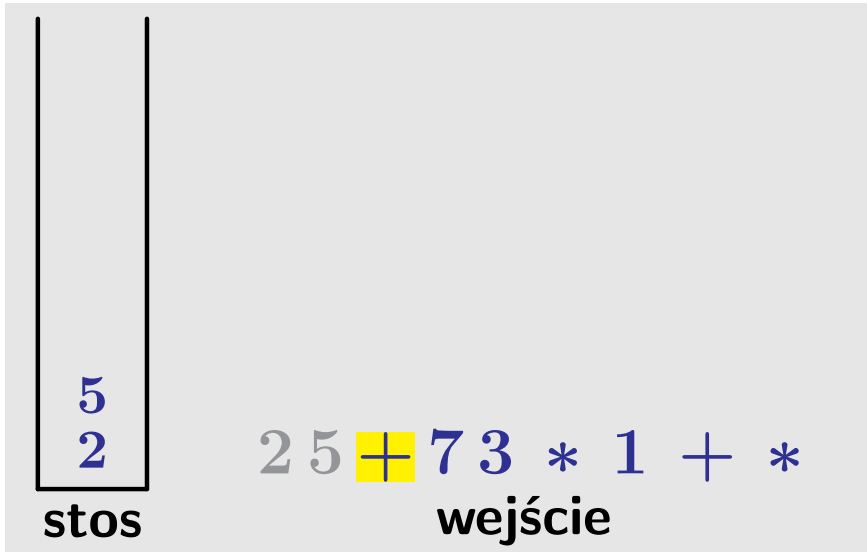


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

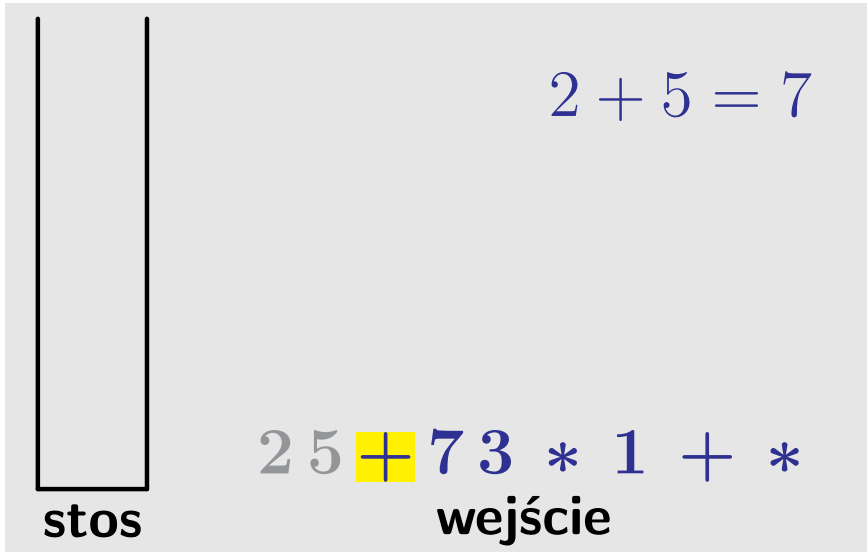


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

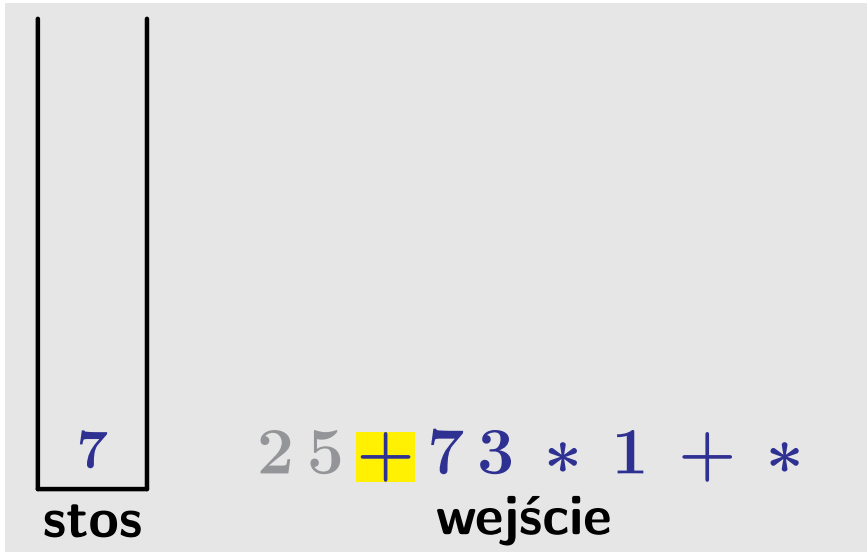


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

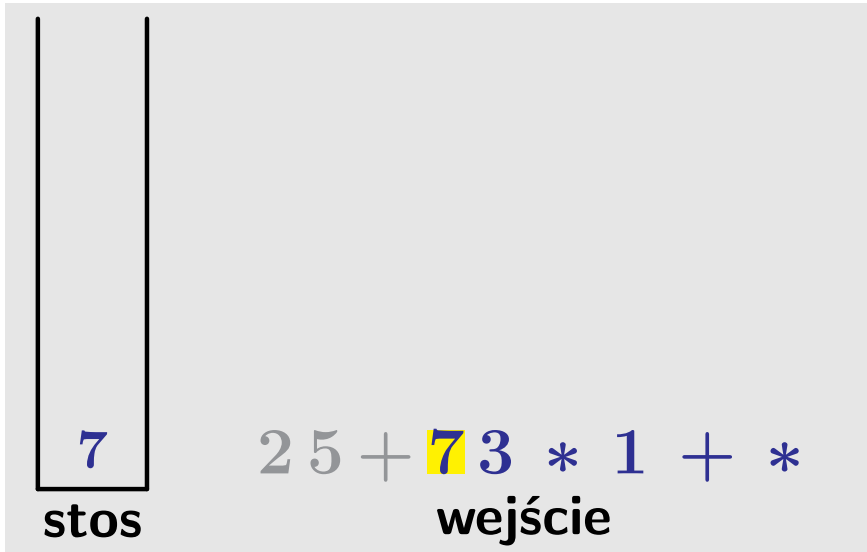


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

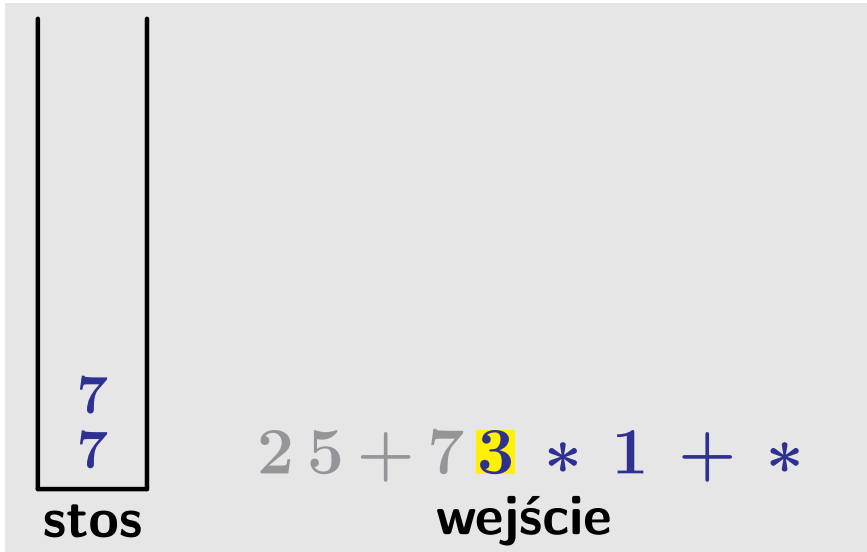


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);



Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

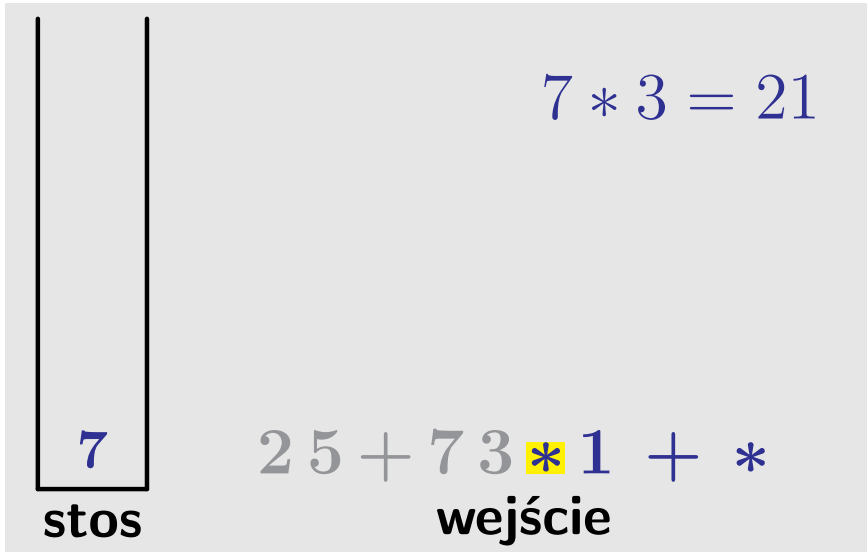
do {
  jeśli na wejściu jest liczba ℓ,
    to przełóż ją z wejścia na stos;
  jeśli na wejściu jest operator *op*,
    to {
      zdejmij ze stosu $\mathbb{A}(op)$ liczb,
      zastosuj do nich operację *op*,
      jej wynik włóż na stos;
    }
} **while** (jest jeszcze coś na wejściu);

Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

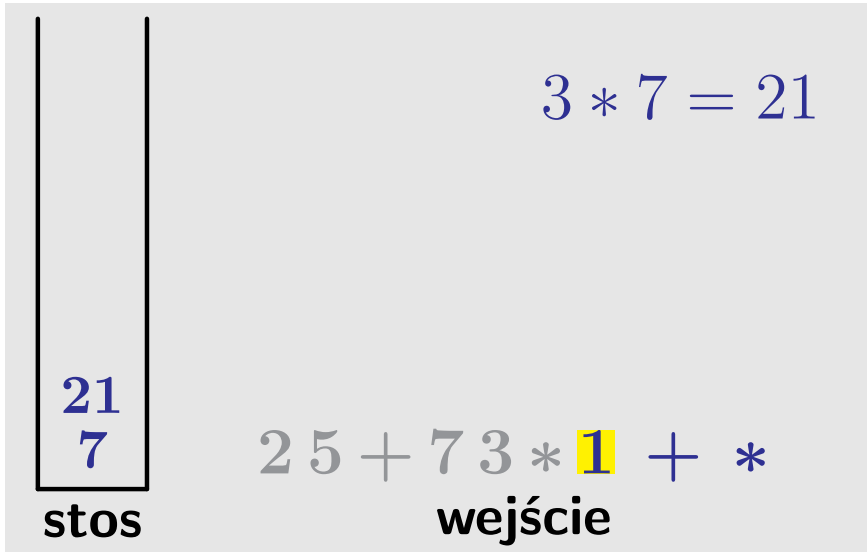


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

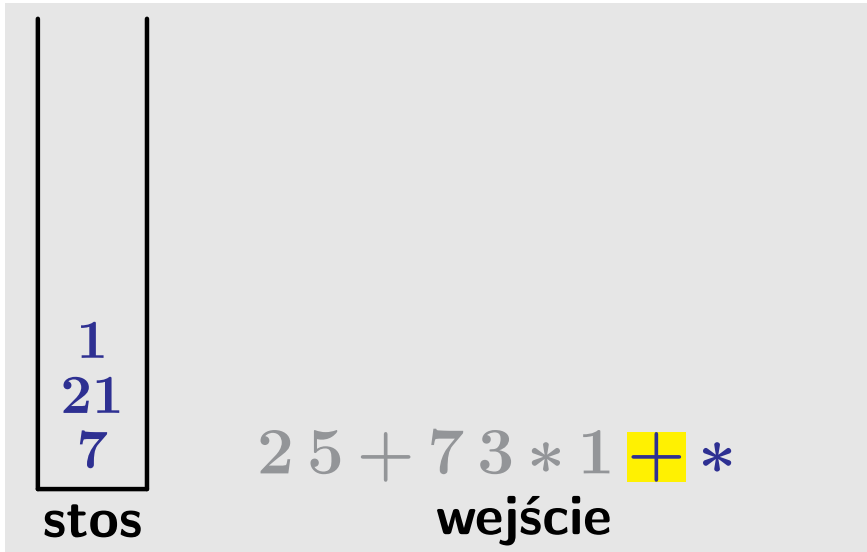


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

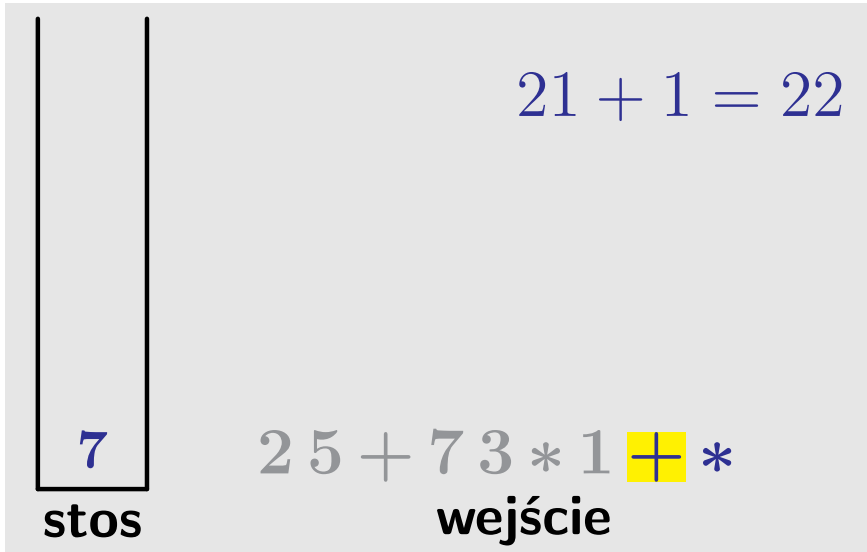


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

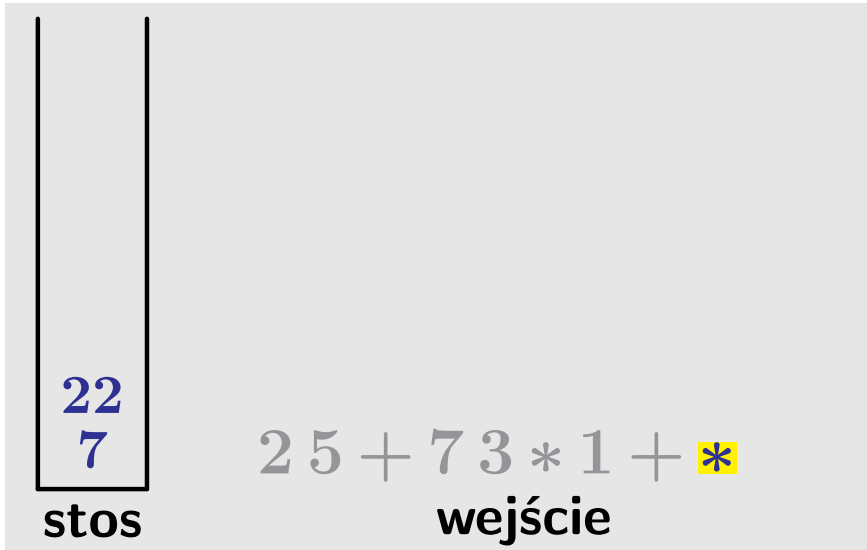


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

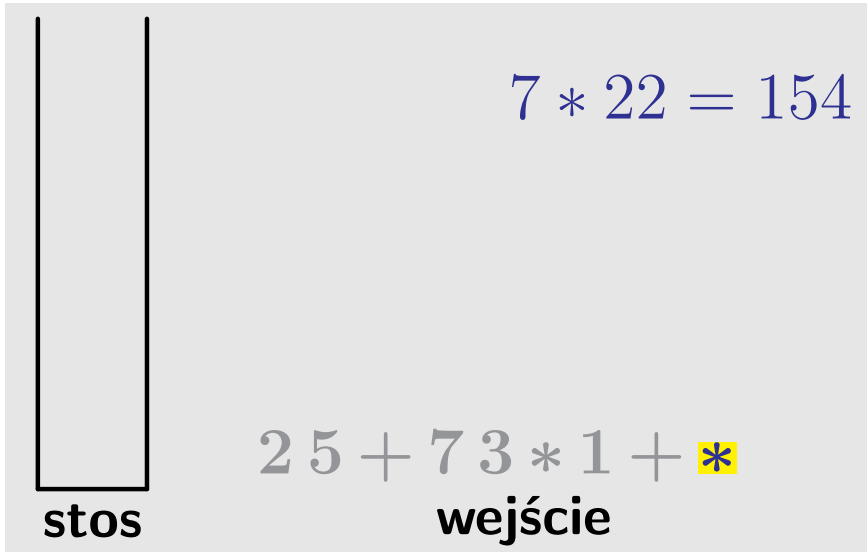


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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);

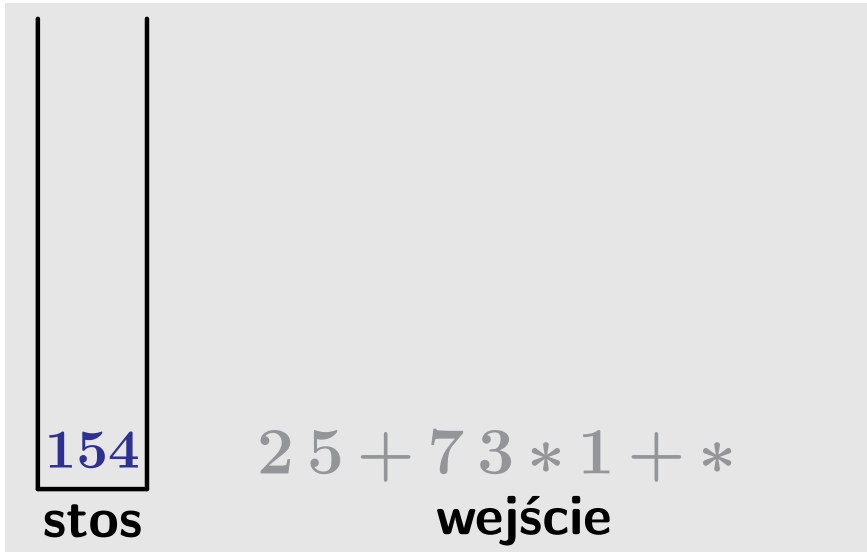


Obliczanie wyrażeń w odwrotnej notacji polskiej

Założenie: każdy operator *op* „wie” ile potrzebuje argumentów: $\mathbb{A}(op)$.
Np. $\mathbb{A}(+) = 2$, $\mathbb{A}(\neg) = 1$, $\mathbb{A}(\texttt{if(...) ... else ...}) = 3$.

Algorytm:

do {
  jeśli na wejściu jest liczba ℓ,
    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

Działania na językach

— takie jak podstawowe konstrukcje wyrażeń regularnych:

1. **Suma:**

$$L \mid M \stackrel{\text{def}}{=} \left\{ w \,\middle|\, w \in L \;\text{ lub }\; w \in M \right\}$$

Działania na językach

— takie jak podstawowe konstrukcje wyrażeń regularnych:

1. **Suma:**

$$L \mid M \stackrel{\text{def}}{=} \left\{ w \,\middle|\, w \in L \;\text{ lub }\; w \in M \right\}$$

2. **Złączenie:**

$$L\,M \stackrel{\text{def}}{=} \left\{ uv \,\middle|\, u \in L \;\text{ i }\; v \in M \right\}$$

Działania na językach

— takie jak podstawowe konstrukcje wyrażeń regularnych:

1. **Suma:**

$$L \mid M \stackrel{\text{def}}{=} \left\{ w \,\middle|\, w \in L \;\text{ lub }\; w \in M \right\}$$

2. **Złączenie:**

$$L\,M \stackrel{\text{def}}{=} \left\{ uv \,\middle|\, u \in L \;\text{ i }\; v \in M \right\}$$

3. **Iteracja (domknięcie Kleenego):**

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \ldots u_n \,\middle|\, u_1, u_2, \ldots, u_n \in L \;\text{ i }\; n \geqslant 0 \right\}$$

Działania na językach

PRZYKŁAD: (suma)

$$L \mid M \stackrel{\text{def}}{=} \left\{ w \,\middle|\, w \in L \text{ lub } w \in M \right\}$$

Działania na językach

PRZYKŁAD: (suma)

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

- $\{\text{nie}, \text{chce}, \text{mi}\} \mid \{\text{się}, \text{uczyć}\} =$

Działania na językach

PRZYKŁAD: (suma)

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

- $\{\text{nie}, \text{chce}, \text{mi}\} \mid \{\text{się}, \text{uczyć}\} = \{\text{nie}, \text{chce}, \text{mi}, \text{się}, \text{uczyć}\}$

Działania na językach

PRZYKŁAD: (suma)

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

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

Działania na językach

PRZYKŁAD: (suma)

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

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

Działania na językach

PRZYKŁAD: (suma)

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

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

PRZYKŁAD: (złączenie)

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

Działania na językach

PRZYKŁAD: (suma)

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

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

PRZYKŁAD: (złączenie)

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

- $\{\text{czerwony}, \text{zielony}\}\ \{\text{dom}, \text{kapelusz}, \text{rower}\} =$

Działania na językach

PRZYKŁAD: (suma)

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

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

PRZYKŁAD: (złączenie)

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

- $\{\text{czerwony}, \text{zielony}\}\ \{\text{dom}, \text{kapelusz}, \text{rower}\} =$
  $\left\{\begin{array}{lll} \text{czerwonydom}, & \text{czerwonykapelusz}, & \text{czerwonyrower}, \\ \text{zielonydom}, & \text{zielonykapelusz}, & \text{zielonyrower} \end{array}\right\}$

Działania na językach

PRZYKŁAD: (suma)

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

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

PRZYKŁAD: (złączenie)

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

- $\{\text{czerwony}, \text{zielony}\}\ \{\text{dom}, \text{kapelusz}, \text{rower}\} =$
  $\left\{\begin{array}{lll} \text{czerwonydom}, & \text{czerwonykapelusz}, & \text{czerwonyrower}, \\ \text{zielonydom}, & \text{zielonykapelusz}, & \text{zielonyrower} \end{array}\right\}$
- $\{\lambda, a, a^2, a^3, \ldots\}\ \{\lambda, b, b^2, b^3, \ldots\} =$

Działania na językach

PRZYKŁAD: (suma)

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

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

PRZYKŁAD: (złączenie)

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

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

Działania na językach

Własności działań

Działania na językach

Własności działań:

- **Łączność:**

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

Działania na językach

Własności działań:

- **Łączność:**

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

- **Przemienność:**

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

  Uwaga: złączenie nie jest przemienne!

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

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

Działania na językach

Podobieństwa do arytmetyki:

- **suma** języków jest jak **dodawanie** liczb

Działania na językach

Podobieństwa do arytmetyki:

- **suma** języków jest jak **dodawanie** liczb,
  język $\emptyset$ jest jak liczba 0

Działania na językach

Podobieństwa do arytmetyki:

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

Działania na językach

Podobieństwa do arytmetyki:

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

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

Działania na językach

Podobieństwa do arytmetyki:

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

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 (chociaż w bardziej skomplikowany sposób).

Działania na językach

Podobieństwa do arytmetyki:

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

PRZYKŁAD:
Rozwiązaniem równania $x = a \cdot x + b$ jest

$$x = b \cdot \frac{1}{1-a}$$

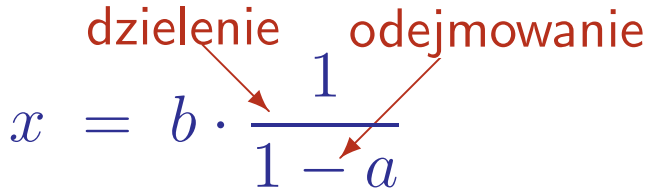
Działania na językach

Podobieństwa do arytmetyki:

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

PRZYKŁAD:

Rozwiązaniem równania $x = a \cdot x + b$ jest



Działania na językach

Podobieństwa do arytmetyki:

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

PRZYKŁAD:

Rozwiązaniem równania $x = a \cdot x + b$ jest

dzielenie odejmowanie

$$x = b \cdot \frac{1}{1-a} = b \cdot \lim_{n\to\infty} \frac{1-a^n}{1-a}$$

jeśli $a < 1$

Działania na językach

Podobieństwa do arytmetyki:

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

PRZYKŁAD:

Rozwiązaniem równania $x = a \cdot x + b$ jest

$$x = b \cdot \frac{1}{1-a} = b \cdot \lim_{n\to\infty} \frac{1-a^n}{1-a} =$$

dzielenie, odejmowanie, jeśli $a < 1$, suma ciągu geometrycznego

Działania na językach

Podobieństwa do arytmetyki:

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

PRZYKŁAD:

Rozwiązaniem równania $x = a \cdot x + b$ jest

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

dzielenie; odejmowanie; jeśli $a < 1$; suma ciągu geometrycznego

Działania na językach

Podobieństwa do arytmetyki:

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

PRZYKŁAD:

Rozwiązaniem równania $x = a \cdot x + b$ jest

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

dzielenie

odejmowanie

jeśli $a < 1$

suma ciągu geometrycznego

Działania na językach

Podobieństwa do arytmetyki:

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

PRZYKŁAD:

Rozwiązaniem równania $x = a \cdot x + b$ jest

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

dzielenie; odejmowanie; jeśli $a < 1$; suma ciągu geometrycznego

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

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

Przykładowe własności:

- $(L \mid M)^* \;=\; (L^* M^*)^* \;=\; (L^* M)^* L^*$

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

Przykładowe własności:

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

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

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:

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

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

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:

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

$$\frac{1}{1-M} \;=\; 1 + \frac{1}{1-M} \cdot M$$

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

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:

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

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

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

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:

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

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

$$\frac{L}{1-M} \;=\; L + \frac{L}{1-M} \cdot M$$

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

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:

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

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

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

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

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:

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

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

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

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\text{ i }\; n \geqslant 0 \right\}$$

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

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:

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

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

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

równanie: $X = L + X \cdot M$ $\longleftrightarrow$ równanie: $X = L \mid XM$

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

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:

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

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

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

równanie: $X = L + X \cdot M$
rozwiązanie: $X = L \cdot \frac{1}{1-M}$

$\longleftrightarrow$

równanie: $X = L \mid XM$

Działania na językach

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

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:

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

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

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

równanie:	$X = L + X \cdot M$	$\longleftrightarrow$	równanie: $X = L \mid XM$
rozwiązanie:	$X = L \cdot \frac{1}{1-M}$		rozwiązanie: $X = LM^*$

Eliminacja lewostronnej rekursji z gramatyki

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

TWIERDZENIE: (rozwiązywanie równania językowego)
Rozwiązaniem równania $X = L \mid XM$ *jest* $X = LM^*$.

Eliminacja lewostronnej rekursji z gramatyki

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

TWIERDZENIE: (rozwiązywanie równania językowego)
Rozwiązaniem równania $X = L \mid XM$ *jest* $X = LM^*$.

lewostronna rekursja

Eliminacja lewostronnej rekursji z gramatyki

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

TWIERDZENIE: (rozwiązywanie równania językowego)
Rozwiązaniem równania $X = L \mid XM$ *jest* $X = LM^*$.

Powyżej:

- X — niewiadoma (szukany język)

Eliminacja lewostronnej rekursji z gramatyki

Iteracja:

$$L^* \stackrel{\text{def}}{=} \left\{ u_1 u_2 \dots u_n \,\middle|\, u_1, u_2, \dots, u_n \in L \;\; \textsf{i} \;\; n \geqslant 0 \right\}$$

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

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

Przekształcenia gramatyk

Interpretujemy

- *nieterminale* jako języki złożone z tych słów, które dają się z nich wyprowadzić

Przekształcenia gramatyk

Interpretujemy

- *nieterminale* jako języki złożone z tych słów, które dają się z nich wyprowadzić;
- *terminale* jako języki 1-literowe

Przekształcenia gramatyk

Interpretujemy

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

Przekształcenia gramatyk

Interpretujemy

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

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

Przekształcenia gramatyk

Interpretujemy

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

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

PRZYKŁAD:

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

Przekształcenia gramatyk

Interpretujemy

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

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

PRZYKŁAD:

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

— *łączność* i *rozdzielność*

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

Przekształcenia gramatyk

Interpretujemy

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

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

PRZYKŁAD:

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

— *łączność* i *rozdzielność*

$$\langle \mathrm{Id} \rangle \ ::= \ \{\mathtt{a} \mid \mathtt{b}\} \mid \langle \mathrm{Id} \rangle \{\mathtt{a} \mid \mathtt{b} \mid \mathtt{0} \mid \mathtt{1}\} \qquad\qquad \langle \mathrm{X} \rangle \ = \ L \mid \langle \mathrm{X} \rangle \, M$$

Przekształcenia gramatyk

Interpretujemy

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

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

PRZYKŁAD:

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

⇓ — *łączność* i *rozdzielność*

$$\langle \mathrm{Id} \rangle \ ::= \ \{\mathtt{a} \mid \mathtt{b}\} \mid \langle \mathrm{Id} \rangle \{\mathtt{a} \mid \mathtt{b} \mid \mathtt{0} \mid \mathtt{1}\} \qquad \langle \mathrm{X} \rangle = L \mid \langle \mathrm{X} \rangle M$$

Przekształcenia gramatyk

Interpretujemy

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

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

PRZYKŁAD:

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

⇓ — *łączność* i *rozdzielność*

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

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

tw. o *rozw. równania językowego* — ⇓

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

Przekształcenia gramatyk

Interpretujemy

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

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

PRZYKŁAD:

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

$\Downarrow$ — *łączność* i *rozdzielność*

$$\langle \mathrm{Id} \rangle ::= \{\mathtt{a} \mid \mathtt{b}\} \mid \langle \mathrm{Id} \rangle \{\mathtt{a} \mid \mathtt{b} \mid \mathtt{0} \mid \mathtt{1}\} \qquad \langle \mathrm{X} \rangle = L \mid \langle \mathrm{X} \rangle M$$

$\Downarrow$ — tw. o *rozw. równania językowego* — $\Downarrow$

$$\langle \mathrm{Id} \rangle ::= \{\mathtt{a} \mid \mathtt{b}\} \{\mathtt{a} \mid \mathtt{b} \mid \mathtt{0} \mid \mathtt{1}\}^* \qquad \langle \mathrm{X} \rangle = L M^*$$