Konwencje funkcyjne

Niech $f : A \times B \to C$.

Aplikacja funkcji na argumenty – klasycznie:

$$f(a, b)$$

Aplikacja funkcji na argumenty – notacja odwrotna:

$$(a, b)f$$

g – funkcja 3 zmiennych.

Klasycznie – $f(g(a, b, c), \sin(x))$,
Odwrotnie – $((a, b, c)g, (x)\sin)f$.

Wyrażenia arytmetyczne

Przykładowe wyrażenie zwykłej notacji arytmetycznej: $1 + 2 \cdot (3 + 4 \cdot 5) - 1 - 1$.

Do zrozumienia takiego wyrażenia potrzeba pewnej „zewnętrznej“ wiedzy:
1. **Priorytety operatorów**.
2. **Łączność**

$a + b + c = (a + b) + c = a + (b + c)$.
$a - b - c = (a - b) - c \neq a - (b - c)$.
$2^{3^4} = 2^{(3^4)} \neq (2^3)^4$.
$a\hat{}b\hat{}c = a\hat{}(b\hat{}c) \neq (a\hat{}b)\hat{}c$.

3. **Nawiasy**

1) i 2) są kwestią konwencji, nie wynikają z samej składni.

Notacja funkcyjna wyrażeń arytmetycznych

Operator arytmetyczny można traktować jako funkcję dwóch argumentów:

$$+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}.$$

Podobnie też:

$$-, \cdot, / : \mathbb{R} \times \mathbb{R} \to \mathbb{R}.$$

(ignorujemy problem dzielenia przez zero)

Wyrażenia arytmetyczne można więc zapisać w notacji funkcyjnej.

wyrażenie	postać funkcyjna
$2 + 2$	$+(2, 2)$
$15 + 2 \cdot a$	$+(15, \cdot(2, a))$
$(1 - 2) \cdot (3/4)$	$\cdot(-(1, 2), /(3, 4))$

Notacja polska

Jak się okaże, zastępując nawiasy i przecinki ustalonym separatorem (np. spacją) nie tracimy jednoznaczności zapisu.
Takie skrócone wyrażenia: *notacja polska* (*NP*, ang. *PN*) wyrażeń.

wyrażenie	postać funkcyjna	wyrażenie NP
$2 + 2$	$+(2, 2)$	$+\ 2\ 2$
$15 + 2 \cdot a$	$+(15, \cdot(2, a))$	$+\ 15\ *\ 2\ \mathrm{a}$
$(1 - 2) \cdot (3/4)$	$\cdot(-(1, 2), /(3, 4))$	$*\ -\ 1\ 2\ /\ 3\ 4.$

Przykład na rekonstrukcję wyrażenia bez nawiasów – tablica.

Nazewnictwo

Dowodzimy jednoznaczności zapisu w NP. Najpierw nazewnictwo:

Token – najmniejsza jednostka składniowa: liczba (również ujemna lub ułamkowa), symbol (stała lub zmienna), nawias, lub operator arytmetyczny (dla uproszczenia tylko $+, -, \cdot, /$).

Wyrażenie (zwykłe lub NP) to ciąg tokenów.

Dla wyrażenia S definiujemy:

- $|S|$ – *długość* S (liczba tokenów w ciągu).
- *Odcinek początkowy* S – ciąg tokenów S' powstający z wykreślenia z S pewnej liczby końcowych tokenów.

Jednoznaczność wyrażeń NP

Twierdzenie

Niech S będzie wyrażeniem NP (ciągiem tokenów). Wtedy:

1. *Jeśli S' jest odcinkiem początkowym S i S' jest wyrażeniem NP, to $S = S'$.*
2. *S jest długości 1 lub istnieje jedyny operator $\circ$ i jedyne wyrażenia NP L i R takie, że*

$$S = \circ\ L\ R$$

(w szczególności istnieje jedyny sposób rekonstrukcji wyrażenia funkcyjnego równoważnego S).

Dowód: (na tablicy) Indukcja względem długości S.

Uwaga: twierdzenie jest prawdziwe dla wyrażeń ogólnych (np. „f g a b c sin x" jest jednoznaczne, o ile znamy liczbę argumentów funkcji f, g, sin).

Odwrotna notacja polska

W NP: operator stawiamy przed argumentami. W *odwrotnej notacji polskiej* (*ONP*, ang. *RPN*): stawiamy je po argumentach (kolejność argumentów nie ulega zmianie).

Wyrażenie (zwykła notacja): $2 + 2$

To samo (odwrotnie, funkcyjnie): $(2, 2)+$

W ONP: $2\ 2\ +$

Odpowiednik Twierdzenia dla NP jest prawdziwy: zapis jest jednoznaczny.

Odwrotna notacja polska

Kilka przykładów:

wyrażenie	postać ONP
$2 + 2$	$2\ 2\ +$
$15 + 2 \cdot a$	$15\ 2\ \mathrm{a}\ *\ +$
$a + b \cdot c$	$\mathrm{a}\ \mathrm{b}\ \mathrm{c}\ *\ +$
$(a + b) \cdot c$	$\mathrm{a}\ \mathrm{b}\ +\ \mathrm{c}\ *$

Kolejność argumentów jest taka, jak w NP (oraz notacji zwykłej).

Jak efektywnie wyznaczać wartość (liczbowego) wyrażenia ONP?
Przykłady na tablicy:
1) $2\ 3\ +\ 4\ 5\ +\ *$
2) $5\ 6$ - 4 * $5\ 2\ 2$ * - -

Odwrotna notacja polska

Przykład ewaluacji – animacja na wykładzie.

Kroki algorytmu ewaluacji (wejście – ciąg tokenów):

1. Stwórz pusty stos.
2. Dla każdego kolejnego tokenu z wejścia:
   - 2a Jeśli tokenem jest liczba: połóż ją na stos.
   - 2b Jeśli tokenem jest operator $\circ$: ściągnij ze stosu prawy argument b , ściągnij ze stosu lewy argument a , wykonaj obliczenie $a \circ b$, połóż wynik na stos.
3. Ściągnij ze stosu i zwróć wynik obliczenia.

Przykładowa implementacja: onp.py w materiałach.

Odwrotna notacja polska

Konwersja wyrażenia zwykłej notacji do ONP: Shunting Yard Algorithm.

Kroki algorytmu konwersji (wejście i wyjście – ciąg tokenów):

1. Stwórz pusty stos operatorów (dolny tor)
2. Dla każdego tokenu z wejścia:
   - 2a. Jeśli tokenem jest liczba lub stała: połóż ten token na wyjście.
   - 2b. Jeśli tokenem jest (: połóż go na stos operatorów.
   - 2c. Jeśli tokenem jest): przesuwaj operatory ze stosu na wyjście, dopóki na szczycie stosu nie pojawi się (; wtedy usuń (ze stosu.
   - 2d. Jeśli tokenem jest operator $\circ$: dopóki na szczycie stosu znajduje się operator o priorytecie wyższym lub równym, niż priorytet $\circ$, ściągaj operatory ze stosu na wyjście; następnie połóż $\circ$ na stosie.
3. Dopóki stos nie jest pusty, ściągaj operatory ze stosu na wyjście.