

Odwrotna notacja polska

Matejko Marek, Mazur Krzysztof, Paszkot Dawid

26 stycznia 2022

Spis treści

1	Wprowadzenie	3
2	Materiały dydaktyczne	4
3	Implementacja	5
4	Dyskusja	6

1 Wprowadzenie

Naszym celem było zapoznanie się z odwrotną notacją polską oraz zrozumieć i ewentualnie poprawić kod zamieszczony w dostarczonym dokumencie.

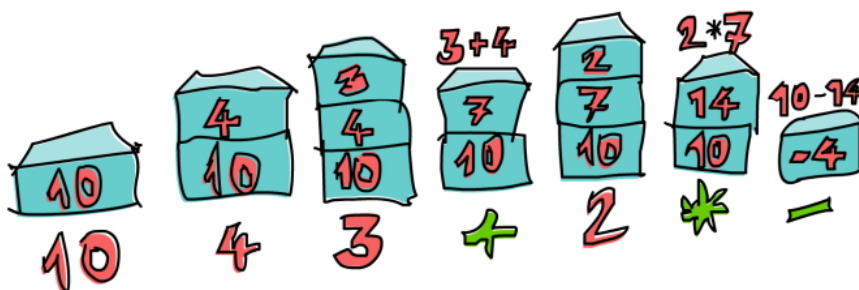
Odwrotna notacja polska ONP (ang. RPN– Reverse Polish Notation) to sposób zapisu wyrażeń arytmetycznych, w którym znak wykonywanej operacji umieszczony jest po operandach, a nie pomiędzy nimi jak w konwencjonalnym zapisie algebraicznym.

Odwrotna notacja polska często jest również zwana notacją Postfix. Została zaproponowana w 1954 r. i wprowadzona we wczesnych latach 60. Powstała na podstawie notacji polskiej, czyli prefixowej, która została stworzona przez polskiego matematyka w 1924 przez Jana Łukasiewicza.

W latach 70. i 80. firma Hewlett-Packard używała Odwrotnej notacji polskiej we wszystkich swoich kalkulatorach biurkowych i ręcznych, a w niektórych modelach używała go do lat 2020. W informatyce odwrotna notacja polska jest używana w językach programowania zorientowanych na stos, takich jak Forth, STOIC, PostScript, RPL i Joy.

2 Materiały dydaktyczne

Aby dowiedzieć się, jak obliczać wyrażenia RPN, pomyślmy o stosie liczb. Przechodzimy przez wyrażenie od lewej do prawej. Za każdym razem, gdy liczba jest napotkana, jest kładzona na wierzch stosu. Kiedy spotykamy operator, bierzemy dwie liczby znajdujące się na szczycie stosu, używamy operatora z tymi dwoma liczbami, a następnie wynik kładziemy na stos. Kiedy dochodzimy do końca wyrażenia, powinna zostać jedna liczba, która reprezentuje wynik (zakładając, że ekspresja była dobrze uformowana).

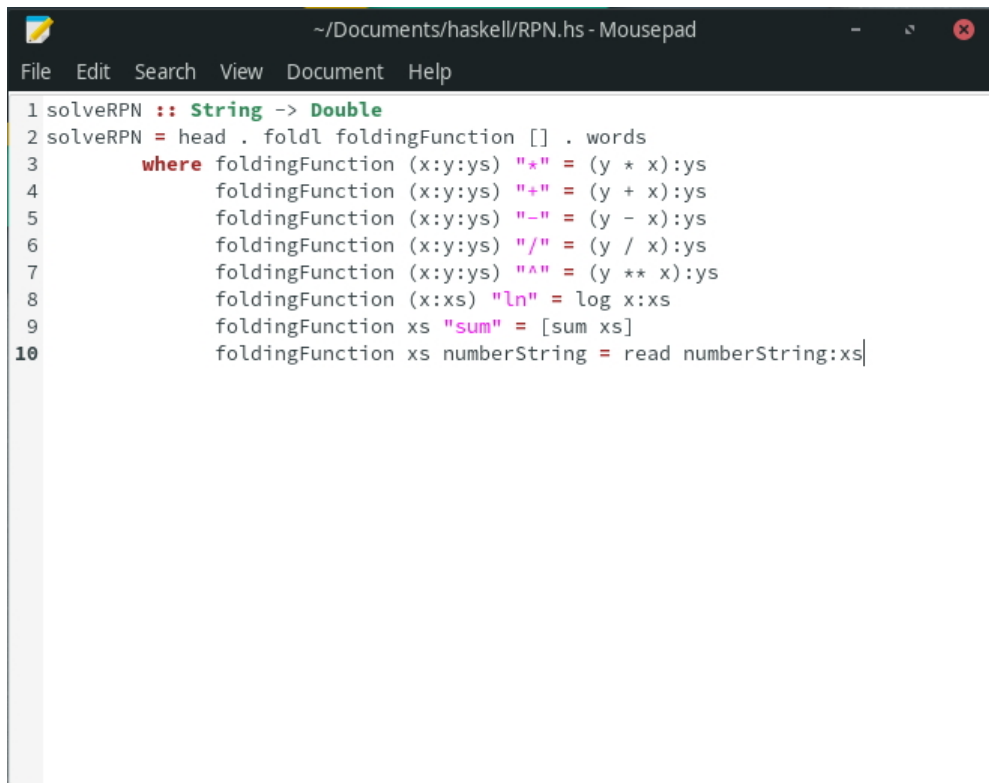


Rysunek 1: Reprezentacja graficzna

Zobaczmy, jak obliczylibyśmy wyrażenie RPN $10\ 4\ 3\ +\ 2\ *\ -$:

1. Wkładamy 10 na stos, więc stos składa się z 10.
2. Następna pozycja to 4, więc ją również wciskamy na stos. Stos jest teraz 10, 4.
3. Robimy to samo z 3, a stos wynosi teraz 10, 4, 3.
4. Spotykamy operator: +. Wyciągamy dwie liczby ze szczytu stosu (więc teraz stos wynosi tylko 10), dodaj te liczby razem i naciśnij ten wynik do stosu. Stos wynosi teraz 10, 7.
5. Odkładamy 2 na stos, więc stos wynosi 10, 7, 2.
6. Spotykamy inny operator. Zrzucamy 7 i 2 ze stosu, mnożymy je i odkładamy wynik na stos. Mnożenie 7 i 2 daje 14, więc stos wynosi teraz 10, 14.

3 Implementacja



```
~/Documents/haskell/RPN.hs - Mousepad
File Edit Search View Document Help
1 solveRPN :: String -> Double
2 solveRPN = head . foldl foldingFunction [] . words
3     where foldingFunction (x:y:ys) "*" = (y * x):ys
4           foldingFunction (x:y:ys) "+" = (y + x):ys
5           foldingFunction (x:y:ys) "-" = (y - x):ys
6           foldingFunction (x:y:ys) "/" = (y / x):ys
7           foldingFunction (x:y:ys) "^" = (y ** x):ys
8           foldingFunction (x:xs) "ln" = log x:xs
9           foldingFunction xs "sum" = [sum xs]
10          foldingFunction xs numberString = read numberString:xs
```

Rysunek 2: Kod źródłowy

Pierwsza linijka kodu to deklaracja typu funkcji, oznacza ona że będziemy brać nasze wpisane liczby jako szereg znaków, czyli string i będziemy zwracać liczbę zmiennoprzecinkową czyli double o podwójnej precyzji.

Druga linijka kodu odpowiada za tworzenie naszego stosu działa na mniej więcej takiej zasadzie że bierzemy nasze wyrażenie i przekształcamy je w listę elementów. Następnie składamy ją w listę przedmiotów za pomocą Foldingfunction. Nasza pusta klamra [] reprezentuje początkowy stos.

Następnie mamy do wyboru którą implementację Foldingfunction wybieremy pierwsze pięć funkcji działają mniej więcej w taki sposób pobierają element z stosu a konkretnie 2 elementy x i y które to są ostatnimi czyli w zasadzie pierwszymi elementami stosu ponieważ dodajemy elementy do stosu od góry i te 2 elementy x i y tworzą naszą nową komurkę.

Następne Foldingfunction to logarytm oraz suma gdzie logarytm działa na naszą pierwszą od góry kolumnę i tylko na nią, natomiast suma łączy wszystkie kolumny w jedną.

Ostatnia linia zawierająca Foldingfunction odpowiada za wpisywanie do stosu liczb jeśli wpiszemy liczbę zostanie ona dodana do naszego stosu.

4 Dyskusja

Udało nam się otrzymać działający kalkulator Odwróconej notacji polskiej zgodnie z postawionym przed nami zadaniem.