

Wojciech Goralewski s464983

Zadanie algorytmiczne nr 5

W algorytmie będę najpierw sprawdzał czy słowo jest puste lub n wynosi 0 – jeżeli tak to zwracam symbol pusty, następnie sprawdzę czy potęga wynosi 1 – wtedy będę zwracał P . Potem w pętli `for` będę stosował operację konkatencji nowej zmiennej P która finalnie będzie wyjściem ze słowem P .

Wejście – P, n

Wyjście – newP

POW(P, n)

`symbol := e`

`newP := []` %inicjalizacja zmiennych%

`if LENGTH(P) == 0 or n == 0 then`

`return symbol` %jeżeli P to słowo puste lub $n=0$ zwróć symbol pusty%

`if n == 1 then`

`return P` % P do potęgi 1 to P %

`for i=2 to n do`

`newP = CON(newP,P)` %od $n=2$ do n łącz nowo powstałe słowo z oryginalnym P %

`return newP`

Zadanie algorytmiczne nr 6

W algorytmie będę najpierw sprawdzał poprawność i długość słów Q i P. Jeżeli Q to słowo puste to jest pod słowem każdego P. Jeżeli P jest słowem pustym, a Q nie jest (bo przeszedł pierwszy IF) to Q nie może być pod słowem. Jeżeli Q jest dłuższym słowem niż P to naturalnie nie może być jego pod słowem.

Następnie w dwóch pętlach będę za pomocą funkcji POS sprawdzał czy w słowie P znajdują się taki podciąg symboli który pasuje do odpowiednich wyrazów Q. W pierwszej pętli wartość końcowa to $LENTH(P) - LENGTH(Q) + 1$ ponieważ w przypadku gdzie słowo Q jest pod słowem na końcu słowa P możemy zaoszczędzić lenQ ilości powtórzeń pierwszej pętli. Innymi słowy tworzę przesuwające się po słowie P okno o rozmiarze długości słowa Q i sprawdzam czy wszystkie aktualne symbole pasują do siebie.

Wejście – Q,P

Wyjście – ‘TAK’ lub ‘NIE’

SUBSTRING(Q,P)

lenP := LENGHT(P)

lenQ := LENGTH(Q) %inicjalizacja zmiennych%

if lenQ == 0 then %jeżeli Q=słowo puste zwróć TAK%

return ‘TAK’

if lenP == 0 or lenQ > lenP then %jeżeli P=słowo puste albo Q jest dłuższe niż P %

return ‘NIE’ %zwróć NIE bo w takiej sytuacji Q nie może być pod słowem%

for i=1 to lenP – lenQ + 1 do

flag := true %zmienna która będzie monitorować czy symbole pasują do siebie%

for j=1 to lenQ do

if POS(Q,j) != POS(P, i - 1 + j) then %jeżeli jakikolwiek symbol się nie zgadza%

flag := false %ustaw na false %

break %przerwij pętle i idź do i+1%

if flag == true then %jeżeli w jakimkolwiek momencie zmienna została%

return ‘TAK’ %niezmieniona zakończ program. Nie ma potrzeby%

%dalszej iteracji słowa%

return ‘NIE’ % zwróć NIE w przypadku gdy zmienna nigdy nie była TRUE%