

Zadanie algorytmiczne nr 13

W algorytmie biorę każde słowo z języka L1 i za pomocą funkcji CON() w pętli while tworzę konkatenację z każdym słowem z L2 i dodaje do nowego języka. Powtarzam tą czynność przez wszystkie wyrazy z L1. Końcowo otrzymam konkatenację dwóch języków. Jeżeli jakkolwiek język będzie pusty to zwracam pustą listę.

Wejście: L1, L2 – dwa języki

Wyjście: conL – konkatenacja języków L1 i L2

CONLANG(L1, L2)

conL := [-]

pomL2 := L2

```
while L1 != [-] do                                %w pierwszej pętli pobieram słowo z L1%
    słowoL1 := HEAD(L1)
    while L2 != [-] do                            % w drugiej pętli pobieram słowo z L2 i konkatenuje z L1%
        słowoL2 = HEAD(L2)
        noweSłowo = CON(słowoL1, słowoL2)
        conL = MAKELIST(noweSłowo, conL)         %nowe słowo dodaje do nowego języka%
        L2 := TAIL(L2)
    L1 := TAIL(L1)    %aktualizacja L1%
    L2 := pomL2      %resetuję język L2, aby na nowo konkatenować z kolejnym słowem z L1%
return conL
```

Zadanie algorytmiczne nr 14

Algorytm najpierw sprawdzi poprawność wpisanego 'n'. Następnie jako, że n-ta potęga języka to n-ta konkatenacja tego języka z samym sobą, dlatego wykorzystam funkcję z zadania 13 - CONLANG() - do otrzymania n-tych potęg języka L.

Wejście: L, n – język oraz liczba naturalna 'n' ≥ 0

Wyjście: powL – n-ta potęga języka L

POWLANG(L,n)

powL = [-]

if n < 0 then

 return 'ERROR: n musi być ≥ 0 ' %n nie może być mniejsze od zera %

if n == 0 then

 return [[]] % dla n=0 zwracam język składający się ze słowa pustego%

if n == 1 then

 return L % dla n=1 zwracam oryginalny język podany na wejściu %

powL := L

for i = 2 to n do

 powL := CONLANG(powL, L) % liczenie n-tych potęg języka %

return powL