

Zadanie algorytmiczne nr 15

Algorytm na początku sprawdzi czy języki są puste. Jeżeli jeden z nich jest pusty to zwracam drugi i na odwrót. Jeżeli języki są sobie równe (co sprawdzam za pomocą EQLANG()) to zwracam jeden z języków na wejściu. Następnie jeżeli podane języki nie są puste i nie są sobie równe to tworzę nową listę L1_U_L2, która początkowo zawiera wszystkie słowa z L1 i sprawdzam za pomocą funkcji MEMBERSHIP() czy słowo z L2 występuje na liście L1. Jeżeli nie to znaczy, że muszę to słowo dodać do języka L1_U_L2. W ten sposób otrzymam sumę tych języków.

Wejście: L1, L2 – dwa języki

Wyjście: L1_U_L2 – suma powyższych języków

UNION(L1,L2)

```
If L1 == [-] then
    return L2                                %sprawdzenie czy języki są puste%

If L2 == [-] then
    return L1

If EQLANG(L1, L2) == 'TAK' then              %jeżeli są takie same to zwróć jeden z nich%
    return L1

L1_U_L2 := L1                                %inicjalizacja L1_U_L2 = L1 %

while L2 != [-] do
    slowoL2 = HEAD(L2)
    if MEMBERSHIP(L1, slowoL2) == 'NIE' then    %jeżeli słowo nie należy do L1%
        L1_U_L2 := MAKELIST(slowoL2, L1_U_L2) %dodaje do języka sumy%
    L2 = TAIL(L2)

Return L1_U_L2
```

Zadanie algorytmiczne nr 16

Algorytm na początku sprawdzi czy języki są puste lub czy są sobie równe. Jeżeli jeden z nich będzie pusty to ich część wspólna jest pusta. Jeżeli są sobie równe to ich część wspólna jest równa L1 lub L2. Następnie w pętli będę sprawdzał czy słowo z L1 zawiera się w L2. Jeżeli tak to dodaje to słowo do nowej listy L1_M_L2. W ten sposób otrzymam część wspólną tych języków.

Wejście: L1, L2 - dwa języki

Wyjście: L1_M_L2 – część wspólna języków podanych na wejściu

MEET(L1,L2)

```
If L1 == [-] then
    return [-]                                %sprawdzam czy języki są puste%
If L2 == [-] then
    return [-]
If EQLANG(L1, L2) == 'TAK' then                %jeżeli są sobie równe zwracam jeden z nich%
    return L1
L1_M_L2 := [-]
while L1 != [-]                                %jeśli słowo z L1 należy do L2 to dodaje do listy%
    slowoL1 := HEAD(L1)
    if MEMBERSHIP(L2, slowoL1) == 'TAK'
        L1_M_L2 := MAKELIST(slowoL1, L1_M_L2)
    L1 = TAIL(L1)
return L1_M_L2
```