Wojciech Goralewski s464983

**Zadanie algorytmiczne nr 17**

Jako, że różnica dwóch języków jest operacją niesymetryczną zakładam, że przy podanych dwóch językach na wejściu L1 i L2 użytkownik chce otrzymać różnicę L1\L2. Tak więc będę w pseudokodzie obliczał to co należy do L1, które nie występują w L2. I mój język na wyjściu L3 będzie zawierał właśnie takie słowa.

Algorytm najpierw będzie sprawdzał czy, któryś z podanych języków L1, L2 jest językiem pustym i zwracał odpowiednie wartości. Jeżeli oba języki nie będą puste to w pętli while dopóki L1 nie będzie puste będę sprawdzał po kolei czy słowo z L1 zawiera się w L2. Jeżeli nie to dodaje je do nowego języka L3 który finalnie da nam słowa które zawierają się w L1 i nie zawierają się w L2.

Wejście: L1, L2 – dwa języki

Wyjście: L3 – różnica dwóch języków podanych na wejściu

**SETMINUS(L1,L2)**

If L1 == [-] then %sprawdzenie czy, któryś język jest pusty%

return [-] %L1 puste to zwróć język pusty%

if L2 == [-] then

return L1 %L2 puste to zwróć język L1%

L3 := [-]

while L1 != [-] do

slowoL1 = HEAD(L1) %jeżeli słowo z L1 nie należy do L2 %

if MEMBERSHIP(L2, slowoL1) == ‘NIE’ then

L3 = MAKELIST(slowoL1, L3) %dodaj słowo do nowego języka L3%

L1 = TAIL(L1)

return L3

**Zadanie algorytmiczne nr 18**

Algorytm najpierw sprawdzi czy język L1 jest pusty. Jeżeli tak to zwracam język pusty. Następnie w pętli while biorę każde słowo z L1 i odwracam je za pomocą funkcji REV() i wrzucam je do nowej listy L2 która ostatecznie będzie odbiciem zwierciadlanym L1.

Wejście: L – skończony język

Wyjście: L2 – zwierciadlane odbicie języka L1

**REVLANG(L)**

if L == [-] %jeżeli język L jest pusty zwróć język pusty%

return [-]

L2 := [-]

while L != [-] do %dopóki L nie jest puste pobieraj następne słowa%

slowoL = HEAD(L)

revSlowoL = REV(slowoL) %zrób odbicie zwierciadlane na tym słowie%

L2 = MAKELIST(revSlowoL, L2) %i dodaj je do nowej listy L2%

L = TAIL(L)

return L2