Wojciech Goralewski s464983

**Zadanie Algorytmiczne nr 11**

W pseudokodzie na początku stworzę nową pustą listę, do której następnie będę dodawał słowa z wejściowej listy L i sprawdzał czy słowo brane z listy L nie znajduje się w nowej liście za pomocą MEMBERSHIP (). Jeżeli się nie znajduje to dodaje to słowo do nowej listy aż do wyczerpania całej listy L w pętli while. W ten sposób pozbędę się wszystkich powtórzeń wszytkich słów z listy wejściowej L. Na końcu zwracam nową listę bez powtórzeń czyli język.

Wejście : L – zadana lista słów

Wyjście : noweL – lista bez powtórzeń (język)

**REMOVEREPETITIONS(L)**

noweL := [] % nowa pusta lista %

while L != [] do

slowoL := HEAD(L) %biorę słowo z wejściowej listy%

If MEMBERSHIP(slowoL, noweL) == ‘NIE’ then % sprawdzam czy jest już na nowej liście %

noweL := MAKELIST(noweL, slowoL) %jeżeli nie to dodaje do nowej listy to słowo%

L := TAIL(L) %aktualizuje wejściową listę%

return noweL %zwracam nową listę bez powtórzeń%

**Zadanie Algorytmiczne nr 12**

W pseudokodzie najpierw sprawdzę długości obu języków. Muszę to zrobić ponieważ jeżeli jedna z list posiadała by te same wyrażenia co druga, ale byłaby językiem o parę słów dłuższym to w zależności od wyboru listy do pętli mogę otrzymać ‘TAK’ lub ‘NIE’. Dlatego stworzę 2 nowe pomocnicze języki odpowiadające oryginałom, które pozwolą mi na policzenie długości bez modyfikowania wejściowych list.

Następnie jeżeli oba języki są tej samej długości będę sprawdzał czy każde słowo z L1 należy do L2 za pomocą MEMBERSHIP. Jeżeli w jakimkolwiek momencie w pseudokodzie nie spełni się powyższych wymagań zwracam ‘NIE’. W przeciwnym wypadku zwrócę ‘TAK’

Wejście : L1, L2 – dwa języki

Wyjście: ‘TAK’, ‘NIE’

**EQLANG(L1,L2)**

lenL1 := 0

lenL2 := 0 %inicjalizacja zmiennych potrzebnych do obliczenia długości języków %

pomL1 := L1

pomL2 := L2

while pomL1 != [-] do

pomL1 := TAIL(pomL1) %obliczenie długości L1%

lenL1 := lenL1 + 1

while pomL2 != [-] do

pomL2 := TAIL(pomL2) %obliczenie długości L2%

lenL2 := lenL2 + 1

if lenL1 != lenL2 then %jeżeli nie są takiej samej długości to nie mogą być równe%

return ‘NIE’

while L1 != [-] do

slowoL1 := HEAD(L1) %sprawdzam czy każde słowo z L1 należy do L2%

if MEMBERSHIP(L2, slowoL1) == ‘NIE’

return ‘NIE’ %zwracam ‘NIE’ jeżeli jakiekolwiek słowo nie należy%

L1 = TAIL(L1) %aktualizuję L1%

return ‘TAK’