

ZAM-41

Język programowania

eol-2

ZAM - 41

Język programowania

EOL - 2

dla ZAM - 41

Opis

Opracował Zespół:

prof. dr Leon Łukaszewicz
mgr Iwona Messner
mgr Jan Walasek
mgr Zdzisława Wrotek

Komitet Redakcyjny: J. Borowiec /red. nacz./, W. Kossakowski, A. Mazurkiewicz, J. Wierzbowski, A. Wiśniewski, W. Wudel /sekr. red./

Adres Redakcji Warszawa, ul. Krzywickiego 34

S p i s r z e c z y

Słowo wstępne	VII-VIII
1. WSTĘP	1-1
1.1. Przeznaczenie języka EOL	1-1
1.2. Ogólna struktura języka	1-2
1.3. Formalizm opisu składni	1-3
1.4. Wersje języka EOL	1-4
1.5. Notacja wartości zmiennej	1-4
2. SYMBOLE PODSTAWOWE	2-1
2.1. Znaki	2-1
2.1.1. Litery	2-1
2.1.2. Cyfry	2-1
2.1.3. Znaki specjalne	2-1
3. ZMIENNE I ICH WARTOŚCI	3-1
3.1. Wejście	3-1
3.2. Wyjście	3-2
3.3. Wyrażenie	3-2
3.3.1. Słowo	3-2
3.3.2. Liczba	3-2
3.3.3. Adres zapisu	3-3
3.4. Pliki	3-3
3.5. Zmienna logiczna	3-3
3.6. Stos adresów rozkazów	3-4
3.7. Konwencja wartości początkowych	3-4

4.	POSTAĆ OGÓLNA ROZKAZÓW	4-1
4.1.	Struktura rozkazu	4-1
4.2.	Argumenty	4-1
4.2.1.	Litera z indeksem	4-2
4.2.1.1.	Sposób czytania	4-2
4.2.1.2.	Sposób dopisywania	4-3
4.2.1.3.	Sposób wykorzystania	4-4
4.2.1.4.	Inne litery	4-5
4.2.2.	Etykieta	4-5
4.2.3.	Symbol klasy znaków	4-5
4.2.4.	Tekst	4-6
4.2.5.	Całkowita	4-6
4.2.6.	Brak argumentu	4-7
4.3.	Oznaczoność wyniku	4-7
5.	ROZKAZY, DEKLARACJE, KOMENTARZE	5-1
5.1.	Przesłania	5-1
5.1.1.	Prześlij	5-1
5.1.2.	Czytaj	5-3
5.1.3.	Pisz	5-4
5.1.4.	Kopiuj	5-4
5.1.5.	Wstaw	5-6
5.1.6.	Pobierz	5-6
5.1.7.	Schowaj	5-7
5.1.8.	Ustaw	5-7
5.1.9.	Zamień	5-7
5.2.	Umieszczenie	5-8
5.3.	Usunięcie	5-8
5.4.	Relacja	5-9
5.4.1.	Relacja arytmetyczna	5-9
5.4.2.	Relacja tekstowa	5-10
5.4.3.	Relacja zapisu	5-12
5.5.	Rozkaz sterujący	5-13
5.5.1.	Skok	5-13
5.5.2.	Skok warunkowy	5-13
5.5.3.	Skok zwrotnicowy	5-14

5.5.4.	Skok powrotny	5-16
5.6.	Rozkaz arytmetyczny	5-16
5.7.	Przekształcenie	5-17
5.7.1.	Przekształcenie na słowo	5-18
5.7.2.	Przekształcenie na liczbę	5-18
5.7.3.	Zbijanie	5-19
5.7.4.	Rozbijanie	5-20
5.8.	Szukanie	5-21
5.8.1.	Szukaj	5-21
5.8.2.	Omiń	5-22
5.9.	Zliczanie	5-22
5.9.1.	Znajdź	5-23
5.9.2.	Szukaj i licz	5-23
5.9.3.	Omiń i licz	5-24
5.10.	Różne	5-25
5.10.1.	Cofnij	5-25
5.10.2.	Pakuj	5-25
5.10.3.	Stop	5-25
5.10.4.	Idź	5-25
5.11.	Deklaracja	5-26
5.11.1.	Zwrotnica	5-26
5.11.2.	Robocze	5-27
5.11.3.	Start programu	5-27
5.12.	Komentarz	5-27
6.	PROGRAMY I SEKCJE	6-1
6.1.	Program	6-1
7.	MAKRODEFINICJE	7-1
7.1.	Odwwołanie do makrodefinicji	7-4
8.	PRZYKŁADY	8-1
DODATEK A. ZBIÓR ZNAKÓW DLA DANYCH		
DODATEK B. UŻYCIĘ W PROGRAMIE SEKCJI W MSAS-ie		
DODATEK C. PRZYPISYWANIE URZĄDZEŃ ZMIENNYM WEJŚCIO- WYM /WYJŚCIOWYM/		
DODATEK D. SYGNALIZACJA BŁĘDÓW I PRZYCZYNY ZAKOŃCZE- NIA PROGRAMU		
DODATEK E. LISTA SŁÓW KLUCZOWYCH		
DODATEK F. SKOROWIDZ TERMINÓW		

Słowo wstępne

EOL jest stosunkowo prostym językiem, służącym do przetwarzania symboli, przedstawionych jako ciągi znaków /sznury/. Zakres zastosowań EOL'u jest dość szeroki, aczkolwiek język ten jest przeznaczony w szczególności do pisania translatorów, czyli programów tłumaczących z jednego języka programowania na drugi. Przykładowo, znaczna część translatora języka COBOL na maszynę ZAM 41 została zaprogramowana w EOL'u. Innym przykładem zastosowań EOL'u może być przekształcanie symboliczne wyrażeń arytmetycznych lub logicznych - z tego typu zastosowań pochodzi nazwa języka - Expression Oriented Language.

Język EOL był już opracowany w kilku kolejnych wersjach.

EOL-1 oraz EOL-2 były opracowane koncepcyjnie w latach 1965/66 w Instytucie Maszyn Matematycznych w Warszawie [1].

EOL-2 został następnie udoskonalony oraz zrealizowany w roku 1967 na maszynie ZAM 41 przez zespół: Zdzisława Wrotek, Jan Walasek i Iwona Messner [2]. Ta właśnie wersja

EOL'u została opisana w niniejszym podręczniku.

EOL-3 opracowany został w roku 1967 na uniwersytecie Illinois [3] i zrealizowany w roku 1968 na maszynach IBM 7094 oraz IBM 360. Stał on się punktem wyjścia dla opracowania języka NUCLEOL [4] oraz prac nad EOL-4 traktowanych jako wspólne przedsięwzięcie Uniwersytetu Illinois oraz Uniwersytetu Colorado.

Nie wątpię, że przedstawiona tutaj wersja EOL-2 znajdzie szersze grono użytkowników w naszym kraju.

Leon Łukaszewicz

Warszawa, 10.12.1969 r.

1. WSTĘP

1.1. Przeznaczenie języka EOL

EOL jest prostym językiem do manipulacji symbolami.

EOL ma na celu ułatwienie pisania programów w zakresie p r z y k ł a d o w o następującej problematyki:

- a/ Tłumaczenie z jednego formalnego języka na drugi, w szczególności w zakresie języków programowania.
- b/ Przekształcanie wyrażeń arytmetycznych lub logicznych, na przykład symboliczne różniczkowanie funkcji.
- c/ Ułatwienie wymiany informacji pomiędzy operatorem a maszyną.

Przy projektowaniu języka EOL starano się uzyskać język o stosunkowo prostej budowie, łatwy do nauczenia się oraz realizacji.

W języku EOL zastosowano wiele idei użytych wcześniej w innych językach do manipulacji symbolami, a w szczególności w językach IPL-V oraz COMIT.

1.2. Ogólna struktura języka

W języku EOL wszystkie programy oraz dane wejściowe i wyjściowe mają postać ciągów, złożonych ze znaków pisarskich. W szczególności każdy program w języku EOL może stanowić dane wejściowe lub być wynikiem działania innego programu w tym języku.

Programy w języku EOL składają się z r o z - k a z ó w i d e k l a r a c j i.

Wykonanie programu polega na kolejnym wykonywaniu r o z k a z ó w zawartych w tym programie. Z budowy programu wynika, który z jego rozkazów ma być wykonany jako pierwszy /patrz 5.11.3/. Po zakończeniu wykonania każdego rozkazu, o ile treść jego nie mówi wyraźnie inaczej, wykonany zostaje rozkaz występujący w najbliższej kolejności jego wypisania w programie, czyli tak zwanej s e k w e n - c j i n o r m a l n e j. Wykonanie każdego rozkazu powoduje z zasady wykonanie prostej czynności jak zmiana wartości zmiennej lub zmiana normalnej sekwencji wykonania rozkazów.

Program w języku EOL składa się z ciągu sekcji. Wewnątrz sekcji rozkazy i deklaracje mogą być grupowane w zależne od siebie lub niezależne procedury. Struktura procedur języka EOL wzorowana jest na PL/1.

Ponadto w języku EOL przewidziana jest możliwość deklarowania makrodefinicji, pozwalających na automatyczną modyfikację programu przed jego wykonaniem /patrz 7/.

1.3. Formalizm opisu składni

W niniejszej publikacji opis składni programu i danych sporządzono, wzorując się na notacji zaproponowanej przez Backus'a i zastosowanej w opisie języka ALGOL.

Notację tę uzupełniono następującymi konwencjami:

$\{a|b\}$ oznacza "dokładnie jeden z symboli a, b"

$[a|b]$ oznacza "żaden lub jeden z symboli a, b"

Powyższe reguły uogólnia się w sposób naturalny na dowolną ilość symboli.

$\{a\}..$ oznacza "dowolny ciąg niepusty symboli, z których każdy jest określony przez a"

$[a]..$ oznacza "dowolny, ewentualnie pusty, ciąg symboli, z których każdy jest określony przez a".

P r z y k ł a d y

$\{A B\}$	może oznaczać	A
$\{A B\}..$	"	" ABA
$[A B]..$	"	" ciąg pusty

W niniejszej publikacji dla zaznaczenia dosłowności stosować będziemy symbol cudzośliwu np. ".".

1.4. Wersje języka EOL

Istnieją dwie wersje językowe EOL-u - polska i angielska. Program może się składać z ciągu sekcji napisanych zarówno w wersji polskiej jak i angielskiej. W obrębie jednej sekcji należy stosować wyłącznie słowa kluczowe polskie bądź też angielskie.

W niniejszym opracowaniu podano postać syntaktyczną poszczególnych jednostek w obu wersjach.

1.5. Notacja wartości zmiennej

W przykładach podanych w niniejszej publikacji celem ilustracji wartości zmiennych stosować będziemy następującą notację pomocniczą:

< nazwa zmiennej > : < wartość zmiennej >

Do opisu wartości zmiennej oprócz symboli podstawowych używane są znaczniki.

<znacznik> ::= - | ° | ^ | v | †

gdzie:

- - znacznik słowa,
- ° - " liczby,
- ^ - " adresu,
- v - " zapisu,
- † - " wskazówki.

Pojęcia: słowo, liczba, adres, zapis i wskazówka zostały opisane w rozdziale 3.

P r z y k ł a d y:

I1 :ABCD

I3 :

E4 : $\bar{A}1\bar{=}3$

P4 : $\bar{\vee}BC\bar{D}^{\circ}9\bar{M}R$

Należy podkreślić, że zapisy powyższego typu mają znaczenie wyłącznie pomocnicze i nie mają żadnego wpływu na przebieg wykonania programu lub strukturę danych.

2. SYMBOLE PODSTAWOWE

Język EOL zbudowany jest z następujących symboli podstawowych:

2.1. Znaki

$\langle \text{znak} \rangle ::= \langle \text{litera} \rangle | \langle \text{cyfra} \rangle | \langle \text{znak specjalny} \rangle$

2.1.1. L i t e r y

$\langle \text{litera} \rangle ::= =A|B....|Z$

2.1.2. C y f r y

$\langle \text{cyfra} \rangle ::= 0|1|2.....|9$

2.1.3. Z n a k i s p e c j a l n e

$\langle \text{znak specjalny} \rangle ::= _ | + | ; | - | , | : | ' | \lambda$

Znaczenie symboli:

a/ λ - oznacza przejście do nowej linii,

b/ $_$ - " spację.

Symbole spacji / $_$ / i przecinka /,/ mogą być zastąpione przez dowolną niepustą kombinację spacji i przecinków.

Każdy znak ";", ":" i "λ" może być otoczony z obu stron przez dowolną kombinację spacji i przecinków.

Należy zauważyć, że zbiór znaków języka EOL jest węższy niż zbiór znaków dla danych. Pełny zbiór znaków dla danych zawarty jest w Dodatku A.

3. ZMIENNE I ICH WARTOŚCI

W programach operujemy na ograniczonej ilości zmiennych, wybranych z następującego zbioru:

16	zmiennych wejścia	I1, I2, ..., I16,
16	"	wyjścia Q1, Q2, ..., Q16,
32	"	wyrażeń E1, E2, ..., E32,
32	"	plików P1, P2, ..., P32,
1	zmienna logiczna	H,
1	"	stosu adresów rozkazów /SAR/.

W trakcie wykonywania programu każdej z powyższych zmiennych przypisywane są różne kolejne wartości, których struktura opisana jest poniżej.

Sposób przypisania zmiennym wejścia lub wyjścia urządzeń wejścia lub wyjścia podano w Dodatku C.

3.1. Wejście

<wartość wejścia> ::= <ciąg znaków>
 <ciąg znaków> ::= [<znak>]. .

P r z y k ł a d y

I1 :JAN ⊔ KOWALSKI

I3 :A = B*(C+D) ⊔ λ

3.2. Wyjście

<wartość wyjścia> ::= <ciąg znaków>

P r z y k ł a d

Q2 :X = ⊔3.1415 ⊔; ⊔END

3.3. Wyrażenie

<wartość wyrażenia> ::= [<składnik>] . .

<składnik> ::= <słowo> | <liczba> | <adres zapisu>

3.3.1. S ł o w o

<słowo> ::= { <znak> } . .

P r z y k ł a d y

ABC

X=A+B ⊔ λ;

38

Słowa służą na ogół do zapisu tekstów.

3.3.2. L i c z b a

<liczba> ::= ° <całkowita>

<całkowita> ::= [+ | -] { <cyfra> } . .

Liczby przedstawiają wielkości wyrażone w układzie dziesiętnym. Bezwzględna wartość liczby nie może być większa niż 8388607.

P r z y k ł a d y

°36 °-48 °0006

3.3.3. A d r e s z a p i s u

$\langle \text{adres zapisu} \rangle ::= \{ \langle \text{znak} \rangle \} . .$

Adresem zapisu jest pewien symbol, określający jednoznacznie położenie tego zapisu w pliku.

P r z y k ł a d y

Wyrażenia mogą przyjmować przykładowo następujące wartości:

E1 : $\bar{X}1 = \bar{A} * (\bar{B} + \bar{GAMMA})$

E7 : $\bar{Z} + \bar{Y} = \bar{X}$

E16 : $^{\circ} 13^{\wedge} \text{POB} \bar{1}$

3.4. Pliki

$\langle \text{wartość pliku} \rangle ::= [\langle \text{zapis} \rangle] . . \uparrow [\langle \text{zapis} \rangle] . .$

$\langle \text{zapis} \rangle ::= \langle \text{wartość wyrażenia} \rangle$

P r z y k ł a d y

P1 : \uparrow

P3 : $\vee \bar{A} \bar{B} C^{\circ} 38 \uparrow$

P6 : $\vee a \vee b \vee c \uparrow a \vee e$

gdzie a, b, ... oznaczają zapisy.

3.5. Zmienna logiczna

$\langle \text{wartość logiczna} \rangle ::= + | -$

P r z y k ł a d y

H : +

H : -

3.6. Stos adresów rozkazów

$\langle \text{wartość stosu adresów} \rangle ::= [\langle \text{adres rozkazu} \rangle]$.
 $\langle \text{adres rozkazu} \rangle ::= \{ \langle \text{znak} \rangle \} . .$

Stos adresów rozkazów służy do zapamiętywania adresów rozkazów typu WYKONAJ /CALL/, które jednoznacznie wyznaczają pozycję tych rozkazów w programie.

Adresy te mogą być następnie wykorzystane przez rozkaz WRÓĆ /RETURN/.

P r z y k ł a d

SAR : ^38^63^18

3.7. Konwencja wartości początkowych

Przyjmuje się, że w momencie rozpoczęcia wykonywania programu zmienna H ma wartość "+", natomiast wszystkie pozostałe zmienne są wyzerowane, to znaczy odpowiadające im wartości są ciągami pustymi.

4. POSTAĆ OGÓLNA ROZKAZÓW

4.1. Struktura rozkazu

Struktura każdego rozkazu jest następująca:

```
<operator>_ [<pierwszy argument> [ ,
  <drugi argument> [ , <trzeci argument> ] ] ]
```

Operator jest słowem kluczowym w postaci ciągu dużych liter np. MOVE, CZYTAJ, WYKONAJ.

Postać i znaczenie różnego typu argumentów opisane są poniżej.

4.2. Argumenty

Argumenty w rozkazach mają jedną z następujących postaci:

- Litera indeksowana
- Etykieta
- Symbol klasy znaków
- Tekst
- Całkowita

W niektórych rozkazach pominięcie ostatniego argumentu może mieć sprecyzowane znaczenie.

Z każdą z podanych powyżej postaci argumentu związane jest jego znaczenie opisane poniżej.

4.2.1. L i t e r a z i n d e k s e m

Argument tego typu ma postać

$\langle \text{litera indeksowana} \rangle ::= \langle \text{litera argumentu} \rangle$
 $\langle \text{indeks} \rangle$

$\langle \text{litera argumentu} \rangle ::= A|B|C|D|E|I|N|P|T|Y|Z|K$

$\langle \text{indeks} \rangle ::= 1|2|\dots|32$

Argument w postaci $\langle \text{litera indeksowana} \rangle$ służy do określenia zmiennej oraz wskazuje na sposób operowania wartością tej zmiennej.

Wielkość $\langle \text{litera argumentu} \rangle$ może określać:

- a/ sposób czytania wartości zmiennej,
- b/ sposób dopisywania do wartości zmiennej,
- c/ sposób wykorzystania wartości zmiennej.

Ponadto w niektórych rozkazach występują litery posiadające jeszcze inne znaczenie.

Wielkość $\langle \text{indeks} \rangle$ jest rzeczywistym indeksem zmiennej. Dla przejrzystości opisu rozkazów wprowadzamy następujące symbole dla oznaczenia indeksów:

$\langle n \rangle ::= \langle \text{indeks} \rangle$

$\langle m \rangle ::= \langle \text{indeks} \rangle$

$\langle k \rangle ::= \langle \text{indeks} \rangle$

4.2.1.1. Sposób czytania

Sposób czytania określony jest przez poszczególne litery następująco:

- A czytanie kolejnych składników początkowych z wyrażenia E z jednoczesnym usuwaniem tych składników,
- B czytanie kolejnych składników początkowych z wyrażenia E bez ich usuwania,
- I czytanie kolejnych znaków z wejścia I, z jednoczesnym ich usuwaniem,
- C czytanie kolejnych zapisów znajdujących się bezpośrednio za wskazówką w pliku P z jednoczesnym usuwaniem każdego przeczytanego zapisu,
- D czytanie kolejnych zapisów znajdujących się bezpośrednio za wskazówką w pliku P połączone z jednoczesnym przeskokiem wskazówki o każdy przeczytany zapis. Przeczytane zapisy nie są usuwane z pliku P.

4.2.1.2. Sposób dopisywania

Sposób dopisywania określony jest przez poszczególne litery następująco:

- A dopisywanie na początek E ciągu składników z odwróceniem ich pierwotnej kolejności,
- B dopisywanie na początek E ciągu składników z zachowaniem ich pierwotnej kolejności,
- I dopisywanie na koniec E ciągu składników z odwróceniem ich pierwotnej kolejności,
- Z dopisywanie na koniec E ciągu składników z zachowaniem ich pierwotnej kolejności,

- Q dopisywanie na końcu Q kolejnych składników wyrażenia E,
- C dopisywanie ciągu zapisów bezpośrednio za wskazówką † w pliku P wraz z jednoczesnym odwróceniem pierwotnego porządku tych zapisów,
- D dopisywanie ciągu zapisów bezpośrednio za wskazówką † w pliku P bez zmiany porządku tych zapisów. Jednocześnie wskazówka † umieszczona zostaje bezpośrednio za ostatnim elementem dopisanego ciągu.

4.2.1.3. Sposób wykorzystania

Przez wykorzystanie składnika rozumiemy jego użycie jako argumentu porównania, argumentu w działaniach arytmetycznych itp.

- A wykorzystanie pierwszego składnika E, a następnie usunięcie tego składnika z wyrażenia,
- B wykorzystanie pierwszego składnika E bez jego usunięcia z wyrażenia,
- T porównanie z dowolnym składnikiem E bez jego usunięcia. Porównanie uznaje się za spełnione, gdy jest spełnione w stosunku do chociaż jednego składnika E,
- E porównanie z całym wyrażeniem E bez zmiany wartości tego wyrażenia lub użycie całego wyrażenia,
- I porównanie z pierwszym znakiem wejścia I bez jego usunięcia.

4.2.1.4. Inne litery

Podane niżej litery mają następujące znaczenie:

- N argument określa wyrażenie, którego pierwszy składnik należy interpretować jako liczbę,
- K argument określa wyrażenie, do którego jako pierwszy składnik dopisuje się liczbę wskazującą położenie elementu,
- P argument odnosi się do pliku.

4.2.2. E t y k i e t a

$\langle \text{etykieta} \rangle ::= \langle \text{identyfikator} \rangle$

$\langle \text{identyfikator} \rangle ::= \langle \text{litera} \rangle [\langle \text{litera} \rangle | \langle \text{cyfra} \rangle]^*$

Długość identyfikatora nie może przekraczać 60 znaków. Etykiety stosowane są w programach do oznaczania rozkazów, zwrotnic i procedur.

4.2.3. S y m b o l k l a s y z n a k ó w

W wielu rozkazach zachodzi potrzeba określenia klasy znaków.

$\langle \text{klasa} \rangle ::= \{ L | D | B | R \} \dots$

Znaczenie powyższych symboli jest następujące:

- L oznacza duże litery,
 D " cyfry,
 B " znak "_",
 R " pozostałe znaki.

Klasy określone przez powyższe symbole są rozłączne, a jednocześnie wyczerpują wszystkie stosowane znaki.

4.2.4. T e k s t

<tekst> ::= <ciąg znaków>

Ze względu na budowę tekstów niektóre znaki mają reprezentację dwuznakową a mianowicie:

*L	o	z	n	a	c	z	a	s	y	n	a	s	y	n	a	l	n	i
*	'	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"
**	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"

Przyjęto również dwuznakową reprezentację dla nietypowych symboli kodu wewnętrznego. Szczegóły na ten temat zawiera Dodatek A. Długość tekstu nie może przekraczać 60 znaków. Sekcja programu może zawierać co najwyżej 128 różnych tekstów więcej niż jednoznakowych.

4.2.5. C a ł k o w i t a

Liczby całkowite są używane w tekstach, jako argumenty rozkazów arytmetycznych, rozkazów porównania i w rozkazie UMIEŚĆ.

Na równi z liczbą całkowitą w podanych rozkazach może wystąpić wielkość POLE /SPACE/.

Wielkość ta określa ilość bloków pamięci bębnowej zajętych przez pliki. Długość bloku wynosi 128 słów.

Sekcja programu nie może zawierać więcej niż 128 różnych liczb, których wartość bezwzględna przekracza 15.

4.2.6. B r a k a r g u m e n t u

W niektórych rozkazach brak ostatniego argumentu posiada ściśle sprecyzowane znaczenie /Patrz 5.1.1./.

4.3. Oznaczoność wyniku

W opisie działania wielu rozkazów przyjmuje się, że argumenty spełniają pewne założenia.

Przykładowo:

W rozkazach arytmetycznych zakłada się, że wielkości określone przez oba argumenty rozkazu są liczbami.

W rozkazach zawierających porównania zakłada się, że obie porównywane wielkości są tego samego typu, to znaczy są to dwa słowa, dwie liczby lub dwa adresy zapisów.

W przypadku, gdy jakikolwiek warunek założony w opisie rozkazu nie jest spełniony, wynik działania tego rozkazu jest nieokreślony.

5. ROZKAZY, DEKLARACJE, KOMENTARZE

$\langle \text{rozkaz} \rangle ::= \langle \text{przesłanie} \rangle | \langle \text{umieszczenie} \rangle |$
 $\langle \text{usunięcie} \rangle | \langle \text{relacja} \rangle |$
 $\langle \text{rozkaz sterujący} \rangle |$
 $\langle \text{rozkaz arytmetyczny} \rangle |$
 $\langle \text{przekształcenie} \rangle | \langle \text{szukanie} \rangle |$
 $\langle \text{zliczanie} \rangle | \langle \text{różne} \rangle$

5.1. Przesłania

$\langle \text{przesłanie} \rangle ::= \langle \text{prześlij} \rangle | \langle \text{czytaj} \rangle | \langle \text{pisz} \rangle$
 $\langle \text{kopiuuj} \rangle | \langle \text{wstaw} \rangle | \langle \text{pobierz} \rangle$
 $\langle \text{schowaj} \rangle | \langle \text{ustaw} \rangle | \langle \text{zamień} \rangle$

5.1.1. P r z e ś l i j

$\langle \text{prześlij} \rangle ::= \{ \text{PRZEŚLIJ} | \text{MOVE} \} _ \{ A | B \} \langle n \rangle,$
 $\{ A | B | Y | Z \} \langle m \rangle, \langle \text{test przesłania} \rangle$
 $\langle \text{test przesłania} \rangle ::= \langle \text{ilość} \rangle | \langle \text{klasa} \rangle | \langle \text{tekst} \rangle$
 $\{ B | T \} \langle k \rangle$
 $\langle \text{ilość} \rangle ::= \langle \text{całkowita} \rangle | N \langle k \rangle$

Przeniesienie kolejnych składników z $E \langle n \rangle$ do $E \langle m \rangle$.

Pierwszy argument określa sposób pobierania składników z $E \langle n \rangle$. Drugi argument określa sposób dopisywania tych składników do $E \langle m \rangle$.

Trzeci argument określa moment zakończenia przeniesień. W szczególności składowa <ilość> określa ilość przeniesionych składników.

Pozostałe składowe trzeciego argumentu określają najbliższy składnik, który nie zostaje przeniesiony i na którym kroki przeniesienia zostają zakończone. Składnik ten jest określony w myśl reguł podanych w rozdziale 4.

Brak argumentu oznacza, że należy przenieść wszystkie składniki.

Jeżeli wszystkie składniki z $E \langle n \rangle$ zostaną przeniesione do $E \langle m \rangle$ zanim nastąpi moment zakończenia przeniesień określony przez trzeci argument, to wykonywanie rozkazu zostaje zakończone, a zmiennej H nadana wartość " - ".

P r z y k ł a d y

Założmy, że

$$E1 : \text{XA}^{-1} = 18^{\circ}3';$$

$$E3 : \text{ALFA}$$

$$E8 : ; =$$

$$H : +$$

Wykonanie wypisanych poniżej rozkazów PRZEŚLIJ powoduje w stosunku do powyższych wartości następujące przekształcenia:

PRZEŚLIJ	A1,A3,2	}
PRZEŚLIJ	A1,A3,'='	
PRZEŚLIJ	A1,A3,R	
PRZEŚLIJ	A1,A3,T8	

$$E1 : = 18^{\circ}3';$$

$$E3 : \text{XALFA}$$

PRZEŚLIJ	B1,B3,2
PRZEŚLIJ	B3,B3

$$E3 : \text{XALFA}$$

$$E3 : \text{ALFAALFA}$$

$$H : +$$

PRZEŚLIJ	A3,Y3,R
----------	---------

$$E3 : \text{ALFA}$$

$$H : -$$

5.1.2. C z y t a j

$\langle \text{czytaj} \rangle ::= \{ \text{CZYTAJ} \mid \text{READ} \} _ I \langle n \rangle ,$
 $\{ A \mid B \mid Y \mid Z \} \langle m \rangle [, \langle \text{test przesłania} \rangle]$

Pobieraj kolejne znaki początkowe z $I \langle n \rangle$,
 utwórz z nich jedno słowo i dopisz do $E \langle m \rangle$.

Znaki pobierane z $I \langle n \rangle$ są jednocześnie z
 tego ciągu usuwane. Porządek znaków w utworzo-
 nym słowie jest zgodny z porządkiem ich pobie-
 rania.

Drugi argument określa sposób dopisania
 utworzonego słowa do $E \langle m \rangle$.

Trzeci argument lub jego brak jest określo-
 ny podobnie jak trzeci argument w rozkazie
 $\langle \text{prześlij} \rangle$ z tą różnicą, że odnosi się nie
 do kolejnych składników wyrażenia, a do ko-
 lejnych znaków w ciągu $I \langle n \rangle$.

W przypadku gdy trzeci argument rozkazu
 określa ciąg znaków, to uwzględniony jest je-
 dynie pierwszy znak.

P r z y k ł a d y

Niech

$I1 : X1=28; \lambda _$

$E3 : \neg PQ$

$H: +$

Wypisane poniżej rozkazy powodują:

CZYTAJ I1,A3,5

I1 : ; $\lambda _$

E3 : $\neg X1=28 \neg PQ$

CZYTAJ I1,Y3,D

I1 : 1=28; $\lambda _$

E3 : $\neg PQ \neg X$

CZYTAJ I1,A3,'.' I1 :
 E3 : ^X1=28;λ_PQ
 H : -

5.1.3. P i s z

$\langle \text{pisz} \rangle ::= \{ \text{PISZ} | \text{WRITE} \} _ \{ \text{A} | \text{B} \} \langle n \rangle, Q \langle m \rangle [, \langle \text{test przesłania} \rangle]$

Pobieraj kolejne słowa z E $\langle n \rangle$ i dopisuj je na końcu Q $\langle m \rangle$ jako następujące po sobie ciągi znaków.

Sposób pobierania składników określony jest przez pierwszy argument. Znaczenie trzeciego argumentu lub jego brak jest takie samo jak w rozkazie $\langle \text{prześlij} \rangle$.

P r z y k ł a d y

Niech

E7 : ^X1 = ^13 _ ^ λ
 Q2 : WYNIK _

Wykonanie poniższych rozkazów powoduje:

PISZ A7 ,Q2	E7 :
	Q2 : WYNIK _ X1 = 13 _ λ
PISZ B7,Q2,1	E7 : ^X1 = ^13 _ ^ λ
	Q2 : WYNIK _ X1

5.1.4. K o p i u j

$\langle \text{kopiuuj} \rangle ::= \{ \text{KOPIUJ} | \text{COPY} \} _ \{ \text{C} | \text{D} \} \langle n \rangle, \{ \text{C} | \text{D} \} \langle m \rangle [, \langle \text{test kopiowania} \rangle]$

$\langle \text{test kopiowania} \rangle ::= \langle \text{ilość} \rangle | \langle \text{tekst} \rangle | \{ \text{B} | \text{T} | \text{E} \} \langle k \rangle$

Przenieś kolejne zapisy z $P \langle n \rangle$ do $P \langle m \rangle$.

Pierwszy argument określa sposób pobierania zapisów z $P \langle n \rangle$. Drugi argument określa sposób dopisywania tych zapisów do $P \langle m \rangle$. Trzeci argument określa moment zakończenia przeniesień. W szczególności składowa $\langle \text{ilość} \rangle$ oznacza ilość przeniesionych zapisów.

Pozostałe składowe trzeciego argumentu określają najbliższy zapis, który nie zostaje przeniesiony i na którym wykonywanie rozkazu zostaje zakończone.

Zapis ten jest określony w myśl reguł podanych w rozdziale 4.

Jeśli trzeci argument jest wymieniony i wszystkie zapisy z $P \langle n \rangle$ zostaną przeniesione do $P \langle m \rangle$ zanim nastąpi moment zakończenia przeniesień określony przez trzeci argument, to wykonanie rozkazu zostaje zakończone, a zmiennej H nadana wartość " - ".

Brak trzeciego argumentu oznacza przeniesienie do $P \langle m \rangle$ wszystkich zapisów występujących w $P \langle n \rangle$ z pozostawieniem wartości H bez zmiany.

P r z y k ł a d y

Przyjmijmy, że

$P1 : v_a | v_b \ v_c \ v_d$

$P3 : v_r | v_s \ v_t$

$H : +$

gdzie a, b, c, \dots przedstawiają pewne zapisy. Podane poniżej rozkazy powodują następujące przekształcenia:

KOPIUJ D1,D3,2

P1 : $\forall a \forall b \forall c \exists d$ P3 : $\forall r \forall b \forall c \exists s \forall t$

H : +

KOPIUJ C1,C3,4

P1 : $\forall a \exists$ P3 : $\forall r \exists d \forall c \forall b \forall s \forall t$

H : -

5.1.5. W s t a w

$$\langle \text{wstaw} \rangle ::= \{ \text{WSTAW} | \text{PUT} \} _ \{ A | B \} \langle n \rangle, \{ C | D \} \langle m \rangle$$

Pobieraj kolejne składniki z $E \langle n \rangle$, utwórz z nich jeden zapis i umieść w $P \langle m \rangle$. Pierwszy argument określa sposób pobierania składników z $E \langle n \rangle$. Drugi argument określa sposób dopisania zapisu do $P \langle m \rangle$.

W przypadku, gdy wyrażenie $E \langle n \rangle$ jest puste, wynikiem działania rozkazu jest jedynie przypisanie H wartości " - ".

5.1.6. P o b i e r z

$$\langle \text{pobierz} \rangle ::= \{ \text{POBIERZ} | \text{GET} \} _ \{ C | D \} \langle n \rangle, \\ \{ A | B | Y | Z \} \langle m \rangle$$

Pobierz z $P \langle n \rangle$ zapis i jego składniki dopisz do $E \langle m \rangle$, w sposób określony przez drugi argument.

Pierwszy argument określa sposób pobrania zapisu z $P \langle n \rangle$. W przypadku, gdy wskazówka znajduje się na końcu pliku wynikiem działania rozkazu jest jedynie przypisanie H wartości " - ".

5.1.7. S c h o w a j

$$\langle \text{schowaj} \rangle ::= \{ \text{SCHOWAJ} | \text{SAVE} \} _ P \langle n \rangle, \\ \{ A | B | Y | Z \} \langle m \rangle$$

Adres zapisu w pliku $P \langle n \rangle$, który występuje bezpośrednio przed wskazówką dopisz do wyrażenia $E \langle m \rangle$ w sposób określony przez drugi argument rozkazu.

5.1.8. U s t a w

$$\langle \text{ustaw} \rangle ::= \{ \text{USTAW} | \text{RESTORE} \} _ \{ A | B \} \langle n \rangle, P \langle m \rangle$$

Pobierz z $E \langle n \rangle$ adres zapisu w sposób określony przez pierwszy argument, a następnie wskazówkę w $P \langle m \rangle$ umieść bezpośrednio po zapisie wskazanym przez ten adres. Jeśli zapis ten został uprzednio usunięty z pliku, to wynik tego rozkazu jest nieokreślony.

5.1.9. Z a m i e ń

$$\langle \text{zamień} \rangle ::= \{ \text{ZAMIEŃ} | \text{EXCHANGE} \} _ \{ E \langle n \rangle, \\ E \langle m \rangle | P \langle n \rangle, P \langle m \rangle \}$$

Zamień wzajemnie pomiędzy sobą wartości wyrażen lub plików, wskazanych przez dwa argumenty rozkazu.

P r z y k ł a d

Niech

$E1 : \text{X1} \text{Q}$

$E2 : \text{X2} \text{3}$

Po wykonaniu poniższego rozkazu mamy

ZAMIEŃ $E1, E2$

$E1 : \text{X2} \text{3}$

$E2 : \text{X1} \text{Q}$

5.2. Umieszczenie

$\langle \text{umieszczenie} \rangle ::= \langle \text{umieść w wyrażeniu} \rangle |$
 $\langle \text{umieść w pliku} \rangle | \langle \text{umieść na wyjściu} \rangle$
 $\langle \text{umieść w wyrażeniu} \rangle ::= \{ \text{UMIEŚĆ} | \text{SET} \} _$
 $\{ \langle \text{całkowita} \rangle | \langle \text{tekst} \rangle \}, \{ A | B | Y | Z \} \langle n \rangle$
 $\langle \text{umieść w pliku} \rangle ::= \{ \text{UMIEŚĆ} | \text{SET} \}$
 $\{ \langle \text{całkowita} \rangle | \langle \text{tekst} \rangle \}, \{ C | D \} \langle n \rangle$
 $\langle \text{umieść na wyjściu} \rangle ::= \{ \text{UMIEŚĆ} | \text{SET} \} _ \langle \text{tekst} \rangle,$
 $Q \langle n \rangle$

Wielkość określoną przez pierwszy argument dopisz do zmiennej określonej przez drugi argument. Argument ten określa jednocześnie sposób dopisania podanej wielkości. Wielkość $\langle \text{całkowita} \rangle$ określa składnik wyrażenia, który jest liczbą o podanej wartości.

Jeżeli wskazaną zmienną jest plik, to podana wielkość dopisana jest jako odrębny zapis, składający się z jednego składnika.

P r z y k ł a d y

Niech

E8 : $_ \text{ALFA} _$

Q3 : X1 =

Wypisane niżej rozkazy powodują

UMIEŚĆ 3846,B8 E8 : *3846 $_ \text{ALFA} _$

UMIEŚĆ 'BETA', Y8 E8 : $_ \text{ALFA} _ _ \text{BETA}$

UMIEŚĆ '*L', Q3 Q3 : X1=λ

5.3. Usunięcie

$\langle \text{usunięcie} \rangle ::= \langle \text{usuń z wyrażenia} \rangle |$
 $\langle \text{usuń z wejścia} \rangle | \langle \text{usuń z pliku} \rangle$
 $\langle \text{usuń z wyrażenia} \rangle ::= \{ \text{USUŃ} | \text{CLEAR} \} _ A \langle n \rangle$
 $[, \langle \text{test przesłania} \rangle]$

$\langle \text{usuń z wejścia} \rangle ::= \{ \text{USUŃ} | \text{CLEAR} \} _ I \langle n \rangle [,$
 $\quad \langle \text{test przesłania} \rangle]$

$\langle \text{usuń z pliku} \rangle ::= \{ \text{SKREŚL} | \text{DELETE} \} _ C \langle n \rangle [,$
 $\quad \langle \text{test kopiowania} \rangle]$

Usuń z wejścia, wyrażenia lub pliku, określonego przez pierwszy argument, elementy w ilości określonej przez drugi argument. Znaczenie drugiego argumentu jest takie samo jak odpowiednio w rozkazach $\langle \text{prześlij} \rangle$, $\langle \text{czytaj} \rangle$ lub $\langle \text{kopiuj} \rangle$.

Przykłady

Niech

E1 : $\bar{X}A\bar{=} \bar{=} 18$

I3 : $\lambda _ _ \text{SKIP}$

P7 : $\forall a \uparrow \forall b \forall c$

Wypisane poniżej rozkazy powodują

USUŃ A1, 1

E1 : $\bar{=} \bar{=} 18$

USUŃ I3, 1

I3 : $_ _ \text{SKIP}$

SKREŚL C7

P7 : $\forall a \uparrow$

5.4. Relacja

$\langle \text{relacja} \rangle ::= \langle \text{relacja arytmetyczna} \rangle |$
 $\quad \langle \text{relacja tekstowa} \rangle |$
 $\quad \langle \text{relacja zapisu} \rangle$

5.4.1. Relacja arytmetyczna

$\langle \text{relacja arytmetyczna} \rangle ::=$
 $\quad \langle \text{relacja arytmetyczna równości} \rangle |$
 $\quad \langle \text{relacja arytmetyczna uporządkowania} \rangle$

$\langle \text{relacja arytmetyczna równości} \rangle ::=$
 $\{RW|EQ\} _ \{A|B|T\} \langle n \rangle, \{ \langle \text{całkowita} \rangle | N \langle m \rangle \}$
 $\langle \text{relacja arytmetyczna uporządkowania} \rangle ::=$
 $\{ \{WK|MN\} | \{GT|LT\} \} _ B \langle n \rangle, \{ \langle \text{całkowita} \rangle | B \langle n \rangle \}$

Jeśli pomiędzy liczbami wskazanymi przez pierwszy i drugi argument zachodzi relacja wskazana przez operator rozkazu, to wartość zmiennej H pozostaw bez zmiany. W przeciwnym przypadku zmiennej H nadaj wartość " - ".

W przypadku użycia pierwszego argumentu w postaci $A \langle n \rangle$ usunięcie argumentu ma miejsce tylko wtedy, gdy relacja jest spełniona.

Znaczenie operatorów relacji arytmetycznych jest następujące:

RW /EQ/ - równy
 WK /GT/ - większy
 MN /LT/ - mniejszy

P r z y k ł a d y

Niech

E8 : °8°3

E9 :

RW A8,8

E8:°3

H :bez zmian

RW A8,10

E8: bez zmian

H : -

WK B9,1

wynik nieokreślony

5.4.2. R e l a c j a t e k s t o w a

$\langle \text{relacja tekstowa} \rangle ::=$
 $\langle \text{relacja tekstowa równości} \rangle |$
 $\langle \text{relacja tekstowa uporządkowania} \rangle$

<relacja tekstowa równości> ::=

{RW|EQ} ⊆ {A|B|T|I} <n>, {<tekst>|<klasa>|{B|T}<m>}

<relacja tekstowa uporządkowania> ::=

{{PP|NS} | {PR | FL}} ⊆ B<n>, {<tekst>|B <n>}

Jeśli pomiędzy słowami wskazanymi przez pierwszy i drugi argument rozkazu zachodzi relacja wskazana przez operator rozkazu, to wartość zmiennej H pozostaw bez zmiany. W przeciwnym przypadku umiennej H nadaj wartość " -".

W przypadku użycia pierwszego argumentu w postaci A <n> usunięcie argumentu ma miejsce tylko wtedy, gdy relacja jest spełniona.

Znaczenie operatorów relacji tekstowych jest następujące:

RW /EQ/ - równy

PP /PR/ - poprzedza,

NS /FL/ - następuje.

W przypadku, gdy pierwszy argument ma postać I <n> w porównaniach uwzględniany jest tylko pierwszy znak drugiego argumentu. W tekstowej relacji równości jako argumenty mogą występować adresy zapisów.

Mówimy, że dwa słowa są równe, jeśli są one identyczne. Mówimy, że słowo X poprzedza słowo Y, jeśli przy porównaniu tych słów kolejno znak po znaku przy pierwszej parze różnych znaków znak słowa X jest wcześniejszy od odpowiadającego mu znaku słowa Y.

Przyjmujemy przy tym, że brak znaku jest wcześniejszy od jakiegokolwiek znaku.

Mówimy, że słowo X następuje po słowie Y, jeśli słowo Y poprzedza słowo X. Uporządkowanie znaków: patrz Dodatek A.

P r z y k ł a d y

Niech

E1 : $\bar{A} \bar{BETA}^{\circ 5}$

E2 : $\bar{Q} \bar{A} \bar{3}$

I7: ALFA

Wykonanie następujących rozkazów powoduje

RW B1, 'A'

H : bez zmian

RW I7, B1

E1: $\bar{A} \bar{BETA}^{\circ 5}$

H : bez zmian

5.4.3. R e l a c j a z a p i s u

$$\langle \text{relacja zapisu} \rangle ::= \{RZ|ER\} \cup P \langle n \rangle ,$$

$$\{ \langle \text{całkowita} \rangle | \langle \text{tekst} \rangle |$$

$$\{B|E|N|T\} \langle m \rangle \}$$

Rozkaz powoduje porównanie pierwszego składnika lub całego zapisu z wartością określoną przez drugi argument. Znaczenie liter 2-go argumentu patrz 4.2.1.3. /Sposób wykorzystania/.

W przypadku równości wartość H pozostaw bez zmian. W przeciwnym przypadku nadaj H wartość " - ".

P r z y k ł a d

Niech

P1: $\bar{v}_a \bar{v}_b | \bar{v} \bar{AB} \bar{TOM}$

E3: $\bar{ROM} \bar{AB}$

Wykonanie następującego rozkazu spowoduje:

RZ P1, T3

H :bez zmian

5.5. Rozkaz sterujący

$\langle \text{rozkaz sterujący} \rangle ::= \langle \text{skok} \rangle | \langle \text{skok warunkowy} \rangle |$
 $\langle \text{skok zwrotnicowy} \rangle |$
 $\langle \text{skok powrotny} \rangle$

5.5.1. S k o k

$\langle \text{skok} \rangle ::= \{ \{ \text{SKOCZ} | \text{WYKONAJ} \} |$
 $\{ \text{GOTO} | \text{CALL} \} \} _ \langle \text{etykieta} \rangle$

Przerwij normalną sekwencję rozkazów i przejdź do wykonania rozkazu oznaczonego etykietą. W przypadku rozkazu z operatorem WYKONAJ /CALL/ adres tego rozkazu umieść na wierzchołku stosu SAR.

5.5.2. S k o k w a r u n k o w y

$\langle \text{skok warunkowy} \rangle ::= \{ \{ \text{SKOMI} | \text{SKOPL} | \text{WYMI} |$
 $\text{WYPL} \} _ \{ \text{GOMI} | \text{GOPL} |$
 $\text{CAMI} | \text{CAPL} \} \} _ \langle \text{etykieta} \rangle$

Jeśli spełnione są warunki wskazane przez operator rozkazu, to przejdź do rozkazu oznaczonego etykietą. W przypadku rozkazów z operatorem WYMI /CAMI/ i WYPL /CAPL/ adres tego rozkazu umieść na wierzchołku stosu SAR. W każdym przypadku zmiennej H nadaj wartość " + ".

Poszczególne operatory rozkazu wskazują następujące warunki:

SKOPL /GOPL/, WYPL /CAPL/	wartość H jest równa "+"
SKOMI /GOMI/, WYMI /CAMI/	wartość H jest równa "-"

P r z y k ł a d

Jeśli H jest równe "-" to rozkaz WYMI ALFA powoduje umieszczenie adresu tego rozkazu na wierzchołku stosu SAR i przejście do miejsca programu oznaczonego etykietą ALFA oraz zmianę wartości H na "+".

5.5.3. S k o k z w r o t n i c o w y

$$\langle \text{skok zwrotnicowy} \rangle ::= \{ \{ \text{SKONA} | \text{SKOIN} | \text{WYNA} | \text{WYIN} \} | \{ \text{GONA} | \text{GOIN} | \text{CANA} | \text{CAIN} \} \} _ \{ A | B \} \langle n \rangle, \\ \langle \text{nazwa zwrotnicy} \rangle$$

Przerwij normalną sekwencję rozkazów i przejdź do rozkazu, którego etykieta określona jest przez pierwszy argument rozkazu oraz znajduje się na liście etykiet zwrotnicy o nazwie podanej jako drugi argument.

W przypadku operatorów WYNA /CANA/ oraz WYIN /CAIN/ adres rozkazu umieść na wierzchołku stosu SAR.

Typ zwrotnicy oraz związany z tym sposób wyboru etykiety określone są przez operator rozkazu w sposób następujący:

SKONA /GONA/, WYNA /CANA/ oznaczają, że rozkaz odnosi się do zwrotnicy typu NAZWA i wskazany składnik wyrażenia $E \langle n \rangle$ zawiera słowo równe wybranej etykiecie bądź tekstowi przypisanemu etykiecie na liście etykiet zwrotnicy.

SKOIN /GOIN/, WYIN /CAIN/ oznaczają, że rozkaz odnosi się do zwrotnicy typu INDEKS i wskazany składnik wyrażenia E <n> jest indeksem wybranej etykiety na liście zwrotnicy. Pierwsza etykieta na liście ma indeks równy jeden.

W przypadku, gdy w zwrotnicy typu NAZWA brak jest wskazanej etykiety bądź tekstu przypisanego etykiecie, wówczas sekwencja rozkazów nie zostaje przerwana, a wartość zmiennej SAR pozostaje bez zmian. Ponadto w przypadku operatora WYNA /CANA/ zmiennej H zostaje przypisana wartość " - ".

W przypadku, gdy pierwszy argument ma postać A <n> jego usunięcie ma miejsce tylko wtedy, gdy szukanie zostanie uwieńczone sukcesem. W przypadku, gdy w zwrotnicy typu INDEKS brak jest wskazanej etykiety, wówczas wynik działania rozkazu jest nieokreślony.

P r z y k ł a d y

Niech będą zadeklarowane zwrotnice:

LX : INDEKS LA, LB, LC

KEY: NAZWA FOR, IF, BEGIN, END

Wartości E1 oraz E3 wynoszą:

E1 : °2~A

E3 : ~BEGIN~REAL~X~;

Wówczas rozkaz

SKOIN B1, LX

jest równoważny rozkazowi

SKOCZ LB

Podobnie rozkaz
SKONA B3, KEY
jest równoważny rozkazowi
SKOCZ BEGIN

5.5.4. Skok powrotny

$\langle \text{skok powrotny} \rangle ::= \{ \text{WRÓĆ} | \text{RETURN} \}$

Przerwij normalną sekwencję rozkazów i przejdź do rozkazu następnego w stosunku do rozkazu, którego adres umieszczony jest na wierzchołku stosu SAR. Jednocześnie usuń ten adres ze stosu.

W przypadku, gdy stos SAR jest pusty, znaczenie rozkazu WRÓĆ /RETURN/ jest nieokreślone.

5.6. Rozkaz arytmetyczny

$\langle \text{rozkaz arytmetyczny} \rangle ::= \{ \{ \text{DODAJ} | \text{ODEJMIJ} | \text{MNÓŻ} | \text{DZIEL} \} | \{ \text{ADD} | \text{SUB} | \text{MULT} | \text{DIV} \} \}$
 $_A \langle n \rangle, \{ \langle \text{całkowita} \rangle | B \langle m \rangle \}$

Wykonaj działanie arytmetyczne określone przez operator na liczbach określonych przez argumenty rozkazu i otrzymany wynik umieść na początku wyrażenia E $\langle n \rangle$ z równoczesnym usunięciem 1-go argumentu.

Znaczenie poszczególnych operatorów jest następujące:

DODAJ /ADD/ Dodaj do siebie dwie liczby, określone przez dwa argumenty rozkazu, a

następnie otrzymaną sumę umieść na początku wyrażenia $E \langle n \rangle$.

MNÓŻ /MULT/ Pomnóż przez siebie dwie liczby, określone przez dwa argumenty rozkazu, a następnie otrzymany iloczyn umieść na początku wyrażenia $E \langle n \rangle$.

ODEJMIJ /SUB/ Od liczby określonej przez pierwszy argument rozkazu odejmij liczbę określoną przez drugi jego argument, a następnie otrzymaną różnicę umieść na początku wyrażenia $E \langle n \rangle$.

DZIEL /DIV/ Liczbę określoną przez pierwszy argument rozkazu podziel przez liczbę określoną przez drugi jego argument. Resztę z tego dzielenia umieść jako początkowy składnik $E \langle n \rangle$, a całkowitą część ilorazu jako drugi składnik tego wyrażenia. Znak reszty jest zawsze równy znakowi dzielnej. W przypadku, gdy liczba określona przez drugi argument jest równa zeru wynik dzielenia jest nieokreślony.

Przykłady

Niech

$E1 : ^\circ 9^{-A}$

$E3 : ^\circ -11^{-B}$

Wykonanie poniższych rozkazów powoduje:

DODAJ A1, B3

$E1 : ^\circ -2^{-A}$

DZIEL A1, B3

$E1 : ^\circ 9^{\circ} 0^{-A}$

5.7. Przekształcenie

$\langle \text{przekształcenie} \rangle ::= \langle \text{przekształcenie na słowo} \rangle$
 $\langle \text{przekształcenie na liczbę} \rangle | \langle \text{zbijanie} \rangle$
 $\langle \text{rozbijanie} \rangle$

5.7.1. Przekształcenie na słowo

$\langle \text{przekształcenie na słowo} \rangle ::= \{ \text{SŁOWO} | \text{WORD} \} _ \{ A | B \} \langle n \rangle$

Liczbę stanowiącą początkowy składnik $E \langle n \rangle$ zamień na słowo stanowiące zapis tej liczby w układzie dziesiętnym i otrzymany rezultat umieść na początku $E \langle n \rangle$.

Pierwsza cyfra słowa jest różna od zera z wyjątkiem przypadku, gdy liczba równa się 0, a w przypadku liczby dodatniej pominięty jest znak " + ".

Znaczenie liter pierwszego argumentu patrz 4.2.1.3. /Sposób wykorzystania/.

Przykłady

Niech

$E3 : ^{\circ}6^{\circ}7$

$E5 : ^{\circ}-11^{\bar{A}}$

$E6 : ^{\bar{B}}A$

$E10 :$

Wypisane poniżej rozkazy powodują:

SŁOWO A3

$E3 : ^{\bar{6}}^{\circ}7$

SŁOWO B5

$E5 : ^{\bar{-}}11^{\bar{A}}$

SŁOWO A6

wynik nieokreślony

SŁOWO B10

wynik nieokreślony

5.7.2. Przekształcenie na liczbę

$\langle \text{przekształcenie na liczbę} \rangle ::= \{ \text{LICZBA} | \text{NUMBER} \} _ \{ A | B \} \langle n \rangle$

Słowo stanowiące początkowy składnik E <n> zamień na liczbę stanowiącą wartość tego słowa w układzie dziesiętnym i otrzymany rezultat umieść na początku E <n>.

Znaczenie liter pierwszego argumentu patrz 4.2.1.3. /Sposób wykorzystania/.

P r z y k ł a d y

Niech

E3 : °11°A

E5 : °-07°B

Wykonanie poniższych rozkazów powoduje

LICZBA A3

E3 : °11°A

LICZBA B5

E5 : °-7°-07°B

5.7.3. Z b i j a n i e

<zbijanie> ::= {ZBIJ|COMPRESS}[_{A|B}]<n>[,
<test przesłania>]

Początkowe składniki E <n> ułoż kolejno jeden za drugim i powstały w ten sposób jeden składnik umieść na początku E <n>. Pierwszy argument rozkazu określa sposób pobierania składników z E <n>. Drugi argument lub jego brak określa ostatni z pobranych składników, analogicznie jak w rozkazie <prześlij>.

Jeśli wszystkie składniki E <n> zostaną pobrane zanim warunek określony przez drugi argument zostanie spełniony, to wykonywanie rozkazu zostaje zakończone, a zmiennej H nadana zostaje wartość " - ".

P r z y k ł a d y

Niech

E2 : $\bar{ALFA} = \bar{X} + \bar{3}$ H : +

Wykonanie poniższych rozkazów powoduje

ZBIJ A2, B

E2 : $\bar{ALFA} = X+3$

ZBIJ A2, ';'

E2 : $\bar{ALFA} = X+3$ H : -

5.7.4. R o z b i j a n i e

$$\langle \text{rozbijanie} \rangle ::= \{ \text{ROZBIJ} | \text{SPLIT} \} _ [A|B] \langle n \rangle [,$$

$$\langle \text{test przesłania} \rangle]$$

Słowo stanowiące pierwszy składnik $E \langle n \rangle$ zamień na ciąg złożony z kolejnych znaków początkowych tego słowa i umieść go na początku $E \langle n \rangle$.

Drugi argument rozkazu lub jego brak ma takie samo znaczenie w stosunku do kolejnych znaków początkowego słowa w $E \langle n \rangle$ jak wyrażenie $\langle \text{test przesłania} \rangle$ lub jego brak w rozkazie $\langle \text{czytaj} \rangle$.

W przypadku gdy całe pierwsze słowo w $E \langle n \rangle$ zostanie rozbite przed spełnieniem warunku określonego przez drugi argument rozkazu, wykonywanie rozkazu zostaje zakończone, a zmiennej H nadana wartość " - ".

P r z y k ł a d y

Niech

E1 : $\bar{AB} = \bar{3} \bar{C}$

Wykonanie poniższych rozkazów powoduje

ROZBIJ A1,1

E1 : $\bar{A} \bar{C}$

ROZBIJ B1,1

E1 : $\bar{A} \bar{AB} = \bar{3} \bar{C}$

ROZBIJ A1, B

E1 : $\bar{A} \bar{B} = \bar{3} \bar{C}$ H: -

5.8. Szukanie

<szukanie> ::= <szukaj>|<omiń>

5.8.1. S z u k a j

<szukaj> ::= {SZUKAJ|SEARCH}_ {A|B}<n>,
<nazwa zwrotnicy>

Jeśli na liście zwrotnicy typu NAZWA o nazwie określonej przez ostatni argument rozkazu znajduje się etykieta /bądź tekst przypisany etykiecie/ określona przez pierwszy argument tego rozkazu, to wartość H pozostaw bez zmiany. W przeciwnym przypadku zmiennej H nadaj wartość "-".

W przypadku gdy pierwszy argument ma postać A<n> usunięcie następuje tylko wtedy, gdy szukanie zostaje uwieńczone sukcesem.

P r z y k ł a d y

Niech dana będzie zwrotnica:

KLUCZ :NAZWA IF, PLUS:'+', MINUS: '-', BEGIN

oraz wartość zmiennych

E3 :~+ ~A

E4 :~MARMUR~BAR

Wypisane poniżej rozkazy powodują:

SZUKAJ A3, KLUCZ

E3 : ~A

H : bez zmian

SZUKAJ A4, KLUCZ

E4 : bez zmian

H : -

5.8.2. O m i ń

`<omiń> ::= OMIŃ|SHIFT ⌋P <n>
 [, <test kopiowania>]`

Przesuwaj wskazówkę w pliku P <n> do przodu aż do momentu określonego przez argument <test kopiowania>. Jeżeli argument ten nie występuje w rozkazie, przesuwaj wskazówkę aż do końca pliku.

Argument <test kopiowania> określa moment zatrzymania się wskazówki w ten sam sposób jak ma to miejsce przy czytaniu pliku bez usuwania zapisów w rozkazie <kopiuuj>. Również w tych samych okolicznościach zmiennej H zostaje nadana wartość "-".

P r z y k ł a d y

Przyjmijmy, że

E3 : ^ALFA ^BETA

E23: ^S

P1 : ^a^b^c^d^e

gdzie d : ^ALFA ° 3

natomiast początkowe składniki w zapisach b oraz c są różne od ^ALFA. Wówczas wykonanie następujących rozkazów powoduje:

OMIŃ P1, B3

P1 : ^a^b^c^d^e

OMIŃ P1, 10

P1 : ^a^b^d^e^

H : -

5.9. Zliczanie

`<zliczanie> ::= <znajdź>|<szukaj i licz>|
 <omiń i licz>`

5.9.1. Z n a j d ź

$$\langle \text{znajdź} \rangle ::= \{ \text{ZNAJDŹ} \mid \text{FIND} \} _ \{ A \mid B \} \langle n \rangle, K \langle m \rangle$$

$$[, \langle \text{test przesłania} \rangle]$$

Jeśli w wyrażeniu określonym przez pierwszy argument rozkazu znajduje się składnik określony przez trzeci argument rozkazu, to wartość H pozostaw bez zmiany, a na początku wyrażenia $E \langle m \rangle$ umieść liczbę określającą ilość składników $E \langle n \rangle$ poprzedzających ten składnik.

Pierwszemu składnikowi tego wyrażenia odpowiada liczba jeden. Jeśli takiego składnika brak, to jedynie zmiennej H nadaj wartość "-",

W przypadku braku trzeciego argumentu na początku $E \langle m \rangle$ umieszczona zostaje liczba wskazująca pełną ilość składników w $E \langle n \rangle$.

P r z y k ł a d y

Przyjmijmy, że

$$E3 : \neg AX \neg AY \neg BX \neg CZ$$

$$E4 : \neg Q$$

Wypisane poniżej rozkazy powodują:

$$\text{ZNAJDŹ } A3, K4, 'BX'$$

$$E3 : \neg BX \neg CZ$$

$$E4 : \neg Q$$

$$\text{ZNAJDŹ } B3, K3$$

$$E3 : \neg Q \neg AX \neg AY \neg BX \neg CZ$$

5.9.2. S z u k a j i l i c z

$$\langle \text{szukaj i licz} \rangle ::= \{ \text{SZL} \mid \text{SEC} \} _ \{ A \mid B \} \langle n \rangle, K \langle m \rangle,$$

$$\langle \text{nazwa zwrotnicy} \rangle$$

Wykonaj operację opisaną w 5.8.1. /Szukaj/.
W przypadku znalezienia określonej wartości

na liście zwrotnicy umieść liczbę określającą jej położenie na liście na początku wyrażenia określonego przez drugi argument. Pierwszej wartości na liście zwrotnicy odpowiada liczba 1,

P r z y k ł a d .

Niech dana będzie zwrotnica

KLUCZ :NAZWA IF, FOR, BEGIN, END

oraz wartość zmiennej

E3 :`FOR`A

Poniższy rozkaz powoduje

SZL B3, K3, KLUCZ

E3 :°2`FOR`A

H : bez zmian

5.9.3. O m i ń i l i c z

<omiń i licz> ::= {OML | SHIC} _P <n>, K <m>
[,<test kopiowania>]

Wykonaj operację opisaną w 5.8.2. /Omiń/. Liczbę określającą ilość ominiętych zapisów umieść na początku wyrażenia określonego przez drugi argument.

P r z y k ł a d

Przyjmijmy, że

E3 :`ALFA`BETA

E23 :`S

P1 :`a^b^c^d^e

gdzie

d :`ALFA`3

i pierwsze składniki zapisów b i c są różne od `ALFA`

Poniższy rozkaz powoduje

OML P1, K23, B3 E3 : ^ALFA^BETA
 E23: ^2^S
 P1 : ^a^b ^c | ^d ^e

5.10. Różne

<różne> ::= <cofnij> | <pakuj> | <stop> | <idź>

5.10.1. C o f n i j

<cofnij> ::= {COFNIJ | RESET} _ P <n>

Wskazówkę w pliku P <n> ustaw przed początkowym zapisem w pliku.

5.10.2. P a k u j

<pakuj> ::= {PAKUJ | PACK}

Rozkaz pozwala na odzyskanie obszarów pamięci zajętych przez zapisy usunięte uprzednio przez program. Rozkaz powoduje ustawienie wskazówek we wszystkich plikach przed ich początkowe zapisy.

Wszystkie adresy wskazówek zapamiętano uprzednio stają się nieaktualne.

5.10.3. S t o p

<stop> ::= {STOP | STOP}

Zatrzymaj wykonywanie programu.

5.10.4. I d ź

<idź> ::= {IDŹ | GOSEC} _ <nazwa sekcji>

<nazwa sekcji> ::= <identyfikator>

Przejdź do wykonania sekcji o podanej nazwie. Sekcja ta może być napisana w języku EOL lub MSAS. W przypadku przejścia do sekcji napisanej w języku EOL, wyrażenia i SAR zostają wyzerowane, wartość pozostałych zmiennych jest zachowywana.

Informacje niezbędne do pisania programów mieszanych MSAS-EOL zawiera Dodatek B.

5.11. Deklaracja

<deklaracja> ::= <zwrotnica> | <robocze> | <start>

5.11.1. Z w r o t n i c a

<zwrotnica> ::= <zwrotnica indeksowa> |
<zwrotnica nazwowa>

<zwrotnica indeksowa> ::= { <nazwa zwrotnicy> : }
.. { INDEKS | INDEX }
<lista zwrotnicy indeksowej>

<lista zwrotnicy indeksowej> ::= <etykieta> [,
[-λ] <etykieta>] ..

<nazwa zwrotnicy> ::= <etykieta>

<zwrotnica nazwowa> ::= { <nazwa zwrotnicy> : }
.. { NAZWA | NAME }
<lista zwrotnicy nazwowej>

<lista zwrotnicy nazwowej> ::= <etykieta>
[: <tekst>] [, [-λ] <etykieta> [: <tekst>]] ..

Ze zwrotnicy indeksowej korzysta się za pomocą rozkazów SKOIN /GOIN/, WYIN /CAIN/, zaś ze zwrotnicy nazwowej za pomocą rozkazów SKONA /GONA/ i WYNA /CANA/. Rozkazy te zostały opisane w 5.5.3.

5.11.2. R o b o c z e

`<robocze> ::= {ROBOCZE | WORKSPACE}_`
`<całkowita>`

Deklaracja ta określa ilość bloków pamięci zarezerwowanej na wyrażenia i SAR. Wielkość bloku wynosi 128 słów.

Brak deklaracji równoważny jest zadeklarowaniu maksymalnej ilości pamięci.

5.11.3. S t a r t p r o g r a m u

`<start> ::= {START | START}`

Deklaracja ta poprzedza rozkaz, który w sekcji programu ma być wykonany jako pierwszy. Brak deklaracji `<start>` w sekcji równoważny jest umieszczeniu tej deklaracji przed pierwszym rozkazem sekcji.

5.12. Komentarz

`<komentarz> ::= /*dowolny ciąg znaków, w którym para znaków "*/" nie występuje*/`

Komentarz nie powoduje wykonania żadnej czynności, a służy jedynie do włączenia odpowiednich objaśnień do programu.

6. PROGRAMY I SEKCJE

6.1. Program

$\langle \text{program} \rangle ::= \{ \langle \text{sekcja} \rangle | \langle \text{sekcja M-SAS} \rangle \}$
 . . KONP_ $\langle \text{nazwa sekcji} \rangle$

Nazwa sekcji po słowie KONP wyznacza sekcję, od której rozpoczyna się wykonywanie programu.

$\langle \text{sekcja} \rangle ::= \{ \text{SEKCJA} | \text{SECTION} \} _ \langle \text{nazwa sekcji} \rangle$
 $\{ \lambda | ; \} \{ \langle \text{zdanie zewnętrzne} \rangle$
 $\{ \lambda | ; \} \} . . \{ \text{KONS} | \text{ENDS} \}$

$\langle \text{nazwa sekcji} \rangle ::= \langle \text{identyfikator} \rangle$

$\langle \text{zdanie zewnętrzne} \rangle ::= \langle \text{zdanie} \rangle$

$\langle \text{zdanie} \rangle ::= [\langle \text{etykieta} \rangle ;] . . \langle \text{rozkaz} \rangle |$
 $\langle \text{deklaracja} \rangle | \langle \text{procedura} \rangle |$
 $\langle \text{komentarz} \rangle$

$\langle \text{procedura} \rangle ::= \{ \langle \text{nazwa procedury} \rangle ; \} . .$
 $\{ \text{PROC} | \text{PROC} \} \{ \lambda | ; \} \{ \langle \text{zdanie} \rangle$
 $\{ \lambda | ; \} \} . . \{ \text{KONIEC} | \text{END} \} \{ \lambda | ; \}$

$\langle \text{nazwa procedury} \rangle ::= \langle \text{etykieta} \rangle$

Definicja procedury jest rekursywna, gdyż w skład jednej procedury może wchodzić inna procedura.

Zakresem etykiety umieszczonej we wnętrzu procedury P jest cała zawartość procedury P pomniejszona o zawartość tych procedur, w których wnętrzu ta sama etykieta występuje ponownie.

Zakresem etykiet zdań zewnętrznych /tzn. etykiet tych rozkazów oraz nazw tych procedur i zwrotnic, które są zdaniami zewnętrznymi/ jest cała sekcja również pomniejszona o zawartość procedur, w których wnętrzu ta sama etykieta występuje ponownie. Zakresem nazwy sekcji jest cały program.

P r z y k ł a d

Przykład różnych zakresów etykiet przedstawiony jest w poniższym programie, złożonym z dwóch procedur zewnętrznych o nazwie A oraz P.

Zakres nazwy procedury zewnętrznej A nie przenosi się do wnętrza procedury B, gdyż we wnętrzu tym umieszczona jest inna etykieta A. Dlatego też rozkaz WYKONAJ A odnosi się do procedury A zawartej w procedurze B.

Zakres nazwy procedury zewnętrznej P rozciąga się na całą zawartość procedur A oraz P, gdyż nazwa ta nigdzie nie jest zadeklarowana powtórnie. Dlatego też oba wypisane powyżej rozkazy WYKONAJ P odnoszą się do tej samej procedury P.

Rozkaz SKOMI B powoduje ewentualne przejście do rozkazu SKOCZ A, umieszczonego w procedurze P.

A : PROC

B : PROC

A : PROC

WYKONAJ A

WYKONAJ P

KONIEC

KONIEC

P : PROC

B : SKOCZ A

WYKONAJ P

SKOMI B

KONIEC

7. MAKRODEFINICJE

```

<sekcja źródłowa> ::= <sekcja>|
                        <sekcja modyfikowana>
<sekcja modyfikowana> ::= {SEKCJA | SECTION}_
    <nazwa sekcji> {λ|;} (<lista makrodefi-
        nicji>){λ|;} <treść sekcji modyfikowanej>
<lista makrodefinicji> ::= {<makrodefinicja>}. .
<makrodefinicja> ::= <nazwa makrodefinicji>_
    <lista parametrów>{λ}
    (<treść makrodefinicji>)
<nazwa makrodefinicji> ::= <identyfikator>
<lista parametrów> ::= <parametr>{[,|λ]}
    <parametr>]. .]
<parametr> ::= <identyfikator>
<treść makrodefinicji> ::=
    [<element makrodefinicji>]. .
<element makrodefinicji> ::= <identyfikator>|
    <parametr>|<identyfikator>'<parametr>|
    <całkowita>|<tekst>|_||:|;λ|
    <odwołanie wewnętrzne>|<komentarz>
<odwołanie wewnętrzne> ::=
    *{<nazwa makrodefinicji>|<parametr>}
    [_<argument wewnętrzny>[,
        <argument wewnętrzny>]. .] {λ|;}

```

$\langle \text{argument wewnętrzny} \rangle ::= \langle \text{identyfikator} \rangle |$
 $\langle \text{parametr} \rangle | \langle \text{identyfikator} \rangle' \langle \text{parametr} \rangle |$
 $\langle \text{całkowita} \rangle | \langle \text{tekst} \rangle$

$\langle \text{treść sekcji modyfikowanej} \rangle ::=$
 $\{ \langle \text{element sekcji modyfikowanej} \rangle \} \dots$

$\langle \text{element sekcji modyfikowanej} \rangle ::=$
 $\langle \text{identyfikator} \rangle | \langle \text{całkowita} \rangle \quad | _ | , |$
 $\langle \text{komentarz} \rangle | \langle \text{tekst} \rangle \quad | : | ; | \lambda | \langle \text{odwołanie} \rangle$

$\langle \text{odwołanie} \rangle ::= * \langle \text{nazwa makrodefinicji} \rangle$
 $[_ \langle \text{argument} \rangle [, \langle \text{argument} \rangle] \dots] \quad \{ \lambda ; \}$

$\langle \text{argument} \rangle ::= \langle \text{identyfikator} \rangle | \langle \text{całkowita} \rangle |$
 $\langle \text{tekst} \rangle$

Makrodefinicje pozwalają na modyfikację sekcji programu przed wykonaniem.

Każda makrodefinicja jest określeniem odcinka programu, w którym mogą występować parametry, lecz tylko te, które zostały wymienione na liście parametrów danej makrodefinicji. Odcinek ten zostaje włączony do programu w tych punktach, gdzie nastąpiło odwołanie do danej makrodefinicji.

W wyniku takich modyfikacji $\langle \text{sekcja modyfikowana} \rangle$ przekształca się w $\langle \text{sekcję} \rangle$, która jest następnie wykonywana.

P r z y k ł a d y

a. WYPISZ PERFORATOR
 (PISZ A1, Q'PERFORATOR)

Nazwą powyższej makrodefinicji jest WYPISZ. Makrodefinicja jest jednoparametrowa. Parametrem jest PERFORATOR.

- b. ABC LA, TEKST, ARG
 (UMIEŚĆ TEKST, A'ARG
 SKOMI LA)

Nazwą powyższej makrodefinicji jest ABC. Makrodefinicja jest trójparametrowa. Parametrami jej są LA, TEKST, ARG.

- c. QX TOM, BX, L
 (USUŃ I'L, D
 *ABC EX, 'STOP' ,5
 SKOCZ TOM)

Nazwą powyższej makrodefinicji jest QX.

Makrodefinicja jest trójparametrowa. Jej parametrami są TOM, EX, L. Makrodefinicja ta zawiera odwołanie do makrodefinicji o nazwie ABC.

- d. PUSTE ()

Powyższa makrodefinicja nosi nazwę PUSTE. Makrodefinicja ta jest bezparametrowa. Jej treść jest pusta. Możliwość stosowania makrodefinicji o pustej treści umożliwia np. odwołania do pewnych makrodefinicji na etapie uruchamiania programu, a po uruchomieniu zamianę treści tych makrodefinicji na puste, co pozwala na uniknięcie usuwania zbędnych odwołań z treści sekcji modyfikowanej.

- e. POMOCNICZA A, B ()

Powyższa makrodefinicja nosi nazwę POMOCNICZA. Jej parametrami są A i B. Treść tej makrodefinicji jest pusta.

7.1. Odwołanie do makrodefinicji

Włączenie makrodefinicji do programu ma miejsce tylko wtedy, gdy spełnione są następujące warunki:

- a. Makrodefinicja o danej nazwie została zdefiniowana na liście makrodefinicji danej sekcji modyfikowanej.
- b. Ilość argumentów odwołania jest identyczna z ilością parametrów makrodefinicji.

W momencie odwołania jest ustanawiana odpowiedniość między parametrami i argumentami w ten sposób, że każdemu parametrowi z listy parametrów makrodefinicji odpowiada argument zajmujący tę samą pozycję na liście argumentów.

Po ustanowieniu odpowiedniości następuje włączenie makrodefinicji. Włączenie makrodefinicji polega na zastąpieniu odwołania do makrodefinicji treścią tejże makrodefinicji z równoczesnym podstawieniem argumentów w miejsce odpowiadających im parametrów. Jeśli odwołanie kończy się średnikiem, to po włączeniu treści makrodefinicji średnik jest usuwany.

W przypadku, gdy element makrodefinicji ma postać < identyfikator >!(parametr) , przy podstawianiu usuwany jest apostrof.

Jeśli odwołanie do makrodefinicji powoduje włączenie takiej treści makrodefinicji, w której znajduje się ponowne odwołanie do makrodefinicji, to wykonywane są następujące czynności:

- a. W przypadku, gdy nazwa makrodefinicji jest parametrem pierwotnej makrodefi-

nicji, to zostaje ona zastąpiona odpowiednim argumentem.

- b. Parametry pierwotnej makrodefinicji, które występują w odwołaniu, zostaną zamienione na odpowiadające im argumenty nadrzędnego odwołania.
- c. Zmodyfikowane zgodnie z p. a i b odwołanie zostaje zastąpione treścią odpowiadającą mu makrodefinicji zgodnie z omówionymi wcześniej regułami.

Z powyższego wynika, że włączanie makrodefinicji powoduje przechodzenie argumentów z zewnętrznych odwołań do wewnętrznych. Makrodefinicje muszą być tak zdefiniowane, żeby po ich włączeniu <sekcja modyfikowana> została przekształcona w poprawną syntaktycznie <sekcję>.

P r z y k ł a d y

- a/ Przytoczymy przykład sekcji modyfikowanej o nazwie

STARTOWA i uproszczony schemat jej przekształcenia w <sekcję>. Przykładem argumentu przenoszonego z zewnętrznego odwołania do wewnętrznego jest tutaj GAMMA.

SEKCJA STARTOWA

(ABC LA, TEKST, ARG

(UMIEŚĆ TEKST, A'ARG

SKOMI LA)

QX TOM, EX, L

(USUŃ I'L, D

*ABC EX, 'STOP', 5

SKOCZ TOM))

ALFA : PROC
*QX BETA, GAMMA, 1
GAMMA: STOP
BETA : CZYTAJ I1, B3, B
SKOCZ ALFA
KONIEC
KONS

Pierwszy etap modyfikacji daje w wyniku:

SEKCJA STARTOWA

ALFA : PROC
USUŃ I1, D
*ABC GAMMA, 'STOP' ,5
SKOCZ BETA
GAMMA: STOP
BETA : CZYTAJ I1, B3, B
SKOCZ ALFA
KONIEC
KONS

Drugi etap modyfikacji daje w wyniku <sekcję> :

SEKCJA STARTOWA

ALFA : PROC
USUŃ I1, D
UMIEŚĆ 'STOP', A5
SKOMI GAMMA
SKOCZ BETA
GAMMA : STOP
BETA : CZYTAJ I1, B3, B
SKOCZ ALFA
KONIEC
KONS

b/ Poniżej przytoczymy przykład sekcji modyfikowanej o nazwie PIERWSZA, w której

makrodefinicja P1 zawiera odwołanie do makrodefinicji, której nazwa jest parametrem formalnym ARA.

Schemat przekształcenia tej <sekcji modyfikowanej> na <sekcję> podano niżej.

SEKCJA PIERWSZA

(P1 B1, ARA, C
 (PRZEŚLIJ A'B1, B'C
 *ARA, 3)
 P2 M
 (PISZ A'M, Q1)
 P3 T
 (CZYTAJ I1, B'T, L))

START : PROC

*P1 4, P3, 7

UMIEŚĆ 'ALFABET', Q1

*P1 2, P2, 8

KONIEC

KONS

Pierwszy etap modyfikacji daje w wyniku:

SEKCJA PIERWSZA

START : PROC

PRZEŚLIJ A4, B7

*P3, 3

UMIEŚĆ 'ALFABET', Q1

PRZEŚLIJ A2, B8

*P2, 3

KONIEC

KONS

Następny etap modyfikacji daje w wyniku
 <sekcję>:

SEKCJA PIERWSZA

START : PROC

PRZEŚLIJ A4, B7

CZYTAJ I1, B3, L

UMIEŚĆ 'ALFABET', Q1

PRZEŚLIJ A2, B8

PISZ A3, Q1

KONIEC

KONS

8. PRZYKŁADY

Niżej przytoczono cztery przykłady programów w języku EOL-2. Opis każdego przykładu składa się z następujących części: opis wejścia, opis wyjścia, tekst programu, objaśnienie programu, przykładowe dane i wyniki.

Numery wierszy wypisane z lewej strony tekstu programu nie należą do programu. Zostały one wprowadzone w celu ułatwienia objaśnienia programu.

Przykład 1

Zadaniem programu SORTOWANIE jest wprowadzenie kolejnych słów tekstu, posortowanie tych słów w kolejności alfabetycznej i wypisanie posortowanych słów.

W e j ś c i e. Wejście programu stanowi dowolny tekst zakończony słowem XXX. Słowem tekstu jest dowolny ciąg literowo-cyfrowych symboli.

Separatorem tekstu jest dowolny ciąg symboli różnych od liter i cyfr.

W y j ś c i e. Na wyjściu program wypisuje ciąg posortowanych słów, przy czym każde słowo jest wypisywane w oddzielnym wierszu. Separatory tekstu wejściowego są pomijane.

Plik jest przeglądany kolejno zapis po zapisie. Przed pobraniem kolejnego zapisu zapamiętuje się jego położenie w wyrażeniu E2. Przegląd odbywa się aż do natrafienia na koniec pliku lub na zapis zawierający słowo późniejsze niż bieżące.

P r o g r a m

- (1) SEKCJA SORTOWANIE
- (2) CZYTAJ: COFNIJ P1 /*WCZYTAJ SLOWO*/
- (3) USUN I1,LD
- (4) CZYTAJ I1,A1,RB
- (5) RW A1,'XXXX'
- (6) SKOPL WYPISZ
- (7) TEST: SCHOWAJ P1,A2 /*WPISANIE SLOWA DO PLIKU P1*/
- (8) POBIERZ D1,B2
- (9) SKOMI TEST2
- (10) PP B1,B2
- (11) SKOPL TEST1
- (12) USUN A2
- (13) SKOCZ TEST
- (14) TEST1: USUN A2,1
- (15) USTAW A2,P1
- (16) TEST2: WSTAW A1,D1
- (17) SKOCZ CZYTAJ
- (18) WYPISZ: UMIESC 'L',Q1 /*WYPISANIE SLOW*/
- (19) POBIERZ C1,A2
- (20) PISZ A2,Q1
- (21) SKOPL WYPISZ
- (22) STOP
- (23) KONS

O b j a ś n i e n i e p r o g r a m u

- (2) Ustawienie wskazówki na początku pliku P1
- (3) - (6) Wczytanie kolejnego słowa i sprawdzenie czy jest to słowo końcowe
- (7) - (13) Odszukanie miejsca wpisania słowa do pliku
- (14) - (17) Wpisanie słowa do pliku
- (18) - (22) Wypisanie posortowanych słów w oddzielnych wierszach

P r z y k ł a d o w e d a n e -

PRZED DAWNYMI, BARDZO DAWNYMI CZASY LUDZIE
 MIESZKALI W JASKINIACH.
 MIAST WÓWCZAS NIE BYŁO. NIE BYŁO ŻADNYCH SKLEPOW.
 NIGDZIE NIE SPRZEDAWANO
 CIASTEK ANI CUKIERKOW. TO BYŁY BARDZO
 ZŁE CZASY.

-XXXX-

W y n i k i

ANI
 BARDZO
 BARDZO
 BYŁO
 BYŁO
 BYŁY
 CIASTEK
 CUKIERKOW
 CZASY
 CZASY
 DAWNYMI
 DAWNYMI
 JASKINIACH
 LUDZIE
 MIAST
 MIESZKALI
 NIE
 NIE
 NIE
 NIGDZIE
 PRZED
 SKLEPOW
 SPRZEDAWANO
 TO
 W
 WÓWCZAS
 ŻADNYCH
 ZŁE

Przykład 2

Zadaniem przykładu DRUKI jest wypełnianie formularzy i wypisywanie wypełnionych formularzy na wyjście.

W e j ś c i e. Dane wejściowe są dwóch typów:

a/ Lista formularzy

Każdy formularz poprzedzony jest [. Bezpośrednio po [występuje nazwa formularza, a za nią po przecinku ciąg parametrów formularza zakończony] . Parametrem formularza jest albo dowolny ciąg znaków różnych od] i =, albo = cyfra.

Formalny opis formularza zdefiniowany jest poniżej:

$$\langle \text{formularz} \rangle ::= [\langle \text{nazwa formularza} \rangle , \{ \langle \text{parametr formularza} \rangle \} ..]$$

Znaki [i] oznaczają [i] występujące w danych do programu

$$\langle \text{nazwa formularza} \rangle ::= \langle \text{dowolny ciąg znaków różnych od , i ;} \rangle$$

$$\langle \text{parametr formularza} \rangle ::= \langle \text{tekst} \rangle | = \langle \text{cyfra} \rangle$$

$$\langle \text{tekst} \rangle ::= \langle \text{dowolny ciąg znaków różnych od] i =} \rangle$$

b/ Lista danych do formularzy

Dane do formularza poprzedzone są *, po której występuje nazwa formularza a dalej po przecinku ciąg argumentów aktualnych oddzielonych przecinkami i zakończony ; .

Argumentem aktualnym jest ciąg dowolnych znaków różnych od ; i , .

Formalny opis danych do formularza jest następujący:

```
<dane do formularza> ::= * <nazwa formularza>
                        [ , <argument aktu-
                          alny> ] ..;
<argument aktualny> ::= <dowolny ciąg zna-
                          ków różnych od ;
                          i , >
```

W y j ś c i e. Wczytanie kolejnych danych do formularza powoduje wyszukanie odpowiedniego formularza, a następnie wypisanie na wyjście trzech znaków nowej linii, a po nich wypełnionej zgodnie z danymi treści formularza. Przy wypełnianiu formularza parametr w postaci ciągu znaków przepisywany jest bezpośrednio na wyjście, natomiast w miejsce parametru postaci =cyfra wypisywany jest odpowiedni argument z listy argumentów.

P r o g r a m

- (1) SEKCJA DRUKI
- (2) UMIESC '##[' , B5
- (3) UMIESC ' ; , ' , B6
- (4) ROZBIJ A5
- (5) ROZBIJ A6
- (6) CZYTAJ:USUN 11, T5
- (7) SKOMI KONIEC
- (8) RW 11, '['
- (9) USUN 11, 1
- (10) SKOMI CZYTAJ1

- (11) CZYTAJ2: CZYTAJ 11, B1, ' , '
 (12) USUN 11, 1
 (13) CZYTAJ 11, Z1, ']'
 (14) SKOMI BLAD
 (15) WSTAW A1, D1
 (16) SKOCZ CZYTAJ
- (17) CZYTAJ1: CZYTAJ 11, Z2, T6
 /*WCZYTANIE DANYCH DO FORMULARZA*/
 (18) SKOMI BLAD
 (19) RW 11, ' , '
 (20) SKOMI TWORZ
 (21) USUN 11, 1
 (22) SKOCZ CZYTAJ1
- (23) TWORZ: PRZESLIJ B2, B7
 /*AKTUALIZACJA FORMULARZY*/
 (24) COFNIJ P1
 (25) OMIN P1, B2
 (26) SKOMI BLAD1
 (27) UMIESC ' *L*L*L' , Q1
 (28) POBIERZ D1, B1
 (29) USUN A1, 1
 (30) ROZBIJ A1
 (31) USUN A2, 1
- (32) T1: PRZESLIJ A1, A1, 1
 /*CZY JEST JESZCZE NIE OPRACOWANA
 CZESC FORMULARZA*/
 (33) SKOPL TWORZ3
- (34) T: OMIN P1
 (35) USUN A2
 (36) USUN A7
 (37) SKOCZ CZYTAJ
- (38) TWORZ3: RW A1, ' = '
 (39) SKOPL TWORZ1
 (40) PISZ A1, Q1, ' = '
 (41) SKOMI T
- (42) TWORZ1: USUN A1, D
 (43) LICZBA A1
 (44) PRZESLIJ B2, A4, N1
 (45) SKOMI BLAD1
 (46) PISZ B4, Q1, 1

- (47) USUN A4
 (48) USUN A1,1
 (49) SKOCZ T1
- (50) KONIEC:STOP
- (51) BŁAD: UMIESC '*LBRAK ; LIUB] W DANYCH',Q1
 (52) STOP
- (53) BŁAD1: UMIESC '*LNIEPOPRAWNE DANE=L-',Q1
 (54) PISZ A7,Q1
 (55) SKOCZ T
- (56) KONS
 (57) KONP DRIKI

O b j a ś n i e n i e p r o g r a m u

- (2) - (5) Załadowanie do E5 znaków początków formularzy i danych, do E6 separatora pozycji w danych i znaku końca danych.
- (6) - (10) Usunięcie znaków poprzedzających początek formularza lub danych. Jeśli nie zostanie napotkany znak * lub [to skok do etykiety KONIEC. Jeśli usuwanie zakończyło się na znaku * następuje skok do czytania danych.
- (11) - (16) Wczytanie formularza do wyrażenia E1 i przesłanie go do pliku P1, a następnie skok do czytania.
- (17) - (23) Wczytanie danych do E2 z równoczesnym usunięciem oddzielających je przecinków.

Przesłanie danych do E7 w celu późniejszej ewentualnej sygnalizacji błędu.

- (24) - (26) Szukanie w pliku P1 formularza mającego nazwę taką jak aktualnie opracowywane dane. W przypadku nie znalezienia takiego - skok do sygnalizacji błędu.
- (27) - (31) Wysłanie na wyjście trzech znaków nowej linii, pobranie formularza do wyrażenia E1, rozbicie go na pojedyncze składniki oraz usunięcie nazwy z wyrażen E1 i E2.
- (32) - (37) Sprawdzenie czy formularz ma jeszcze jakieś składniki, jeśli tak to przejście do opracowywania ich, jeśli nie - ustawienie wskazówki na końcu pliku P1, wyczyszczenie wyrażen E2 i E7 i przejście do ponownego czytania.
- (38) - (49) Sprawdzenie typu bieżącego parametru formularza. Jeśli parametr jest tekstem, to wypisanie go na wyjściu, jeśli jest typu=liczba - to wybranie z E2 odpowiadającego mu argumentu aktualnego i wypisanie go na wyjście. Jeśli argument aktualny nie zostanie znaleziony - skok do błędu.
- (50) - (55) Zakończenie działania programu. Sygnalizacja błędu:
- 1/ w przypadku braku ; lub] w danych - zaszyfrowanie błędu i zakończenie działania programu;

2/ w pozostałych przypadkach -
zasygnalizowanie błędu i
przejsście do opracowywania
ewentualnych danych.

Przykładowe dane

[WEZWANIE, OB=1=2 PROSZE ZGŁOSIĆ SIĘ =3=4]
[ZYCZENIA, WSZYSTKIEGO DOBREGO=1=2=3=4]

*ZYCZENIA, DROGIEJ, HANI,
ZYCZA,
KOLEDZY;

*WEZWANIE, JAN, KOWALSKI,
DO WRN W WARSZAWIE,
W DNIU 5.10.1970R. GODZ.12.00.;

*WEZWANIE, ADAM,
DO WRN W WARSZAWIE,
W DNIU 7.10BR.;

*ZYCZENIA,
Z OKAZJI, 25-LECIA PRACY ZAWODOWEJ,
ZYCZY DYREKCJA I WSPOLPRACOWNICY,

;
*WEZWANIE, ANNA, KOWALSKA,
W MHD, -SKLEP NR.2.

;

Wyniki

WSZYSTKIEGO DOBREGO DROGIEJ HANI
ZYCZA
KOLEDZY

OB JAN KOWALSKI PROSZE ZGŁOSIĆ SIĘ
DO WRN W WARSZAWIE
W DNIU 5.10.1970R. GODZ.12.00.

OB ADAM
DO WRN W WARSZAWIE PROSZE ZGŁOSIĆ SIĘ
W DNIU 7.10BR.
NIEPOPRAWNE DANE
-WEZWANIE ADAM
DO WRN W WARSZAWIE
W DNIU 7.10BR.

WSZYSTKIEGO DOBREGO
Z OKAZJI 25-LECIA PRACY ZAWODOWEJ
ZYCZY DYREKCJA I WSPOLPRACOWNICY

OB ANNA KOWALSKA PROSZE ZGLOSIC SIE
W MHD-SKLEP NR.2.

Przykład 3

Zadaniem tego programu jest obliczanie wartości prostych formuł arytmetycznych.

W e j ś c i e. Wejściem programu jest lista formuł arytmetycznych złożonych z liczb całkowitych, nawiasów i znaków czterech działań arytmetycznych.

Każda z formuł znajduje się w oddzielnym wierszu. Znaki spacji i powrotu karetki są nieznaczące. Nie mogą jednak występować pomiędzy cyframi całkowitej. Wiersz, którego pierwszym symbolem znaczącym jest \diamond /romb/ określa koniec danych.

W y j ś c i e. Na wyjściu program wypisuje formułę, znak równości i wartości formuły. Jeżeli formuła jest błędna, to zamiast wyniku program podaje miejsce, w którym wystąpił błąd.

P r o g r a m

- (1) SEKCJA ARYT
- (2) /*DRUKOWANIE TYTULU*/
- (3) UMIESC 'WARTOSCI FORMUL*L',Q1
- (4) /*WCZYTANIE WIERSZA*/

- (5) WCZYT: USUN 11,RDL
 (6) RW 11,'*L'
 (7) SKOPL USPOCZ
 (8) RW 11,'*G'
 (9) SKOMI KLASA
- (10) USPOCZ: USUN 11,1
 (11) SKOCZ WCZYT
- (12) KLASA: RW 11,D
 (13) SKOMI ROZNE
 (14) CZYTAJ 11,Z1,BLR
 (15) SKOCZ USWEWN
- (16) ROZNE: CZYTAJ 11;Z1,1
- (17) USWEWN: USUN 11,LRD
 (18) RW 11,'*G'
 (19) SKOMI LTEST
 (20) USUN 11,1
 (21) SKOCZ USWEWN
- (22) LTEST: RW 11,'*L'
 (23) SKOMI KLASA
- (24) /*BADANIE CZY KONIEC DANYCH*/
- (25) RW A1,'o'
 (26) SKOPL KODAN
- (27) /*LICZENIE I DRUKOWANIE WARTOSCI*/
- (28) PRZESLIJ B1,B2
 (29) UMIESC '*L',Z1
 (30) WYKONAJ FORMULA
 (31) RW A1,'*L'
 (32) SKOMI BLAD
 (33) PISZ A2,Q1
 (34) UMIESC '=',Q1
 (35) SLOWO A3
 (36) PISZ A3,Q1
 (37) UMIESC '*L',Q1
 (38) SKOCZ WCZYT
- (39) /*LICZENIE FORMULY*/
- (40) FORMULA: PROC
 (41) WYKONAJ TERM
- (42) OPERATOR: NAZWA PLUS:'+',MINUS:'-'
- (43) W1: SKONA A1,OPERATOR
 (44) WROC

- (45) PLUS: WYKONAJ TERM
 (46) PRZESLIJ A3,B4,1
 (47) DODAJ A3,B4
 (48) USUN A4,1
 (49) SKOCZ W1
- (50) MINUS: WYKONAJ TERM
 (51) PRZESLIJ A3,B4,1
 (52) ODEJMIJ A3,B4
 (53) USUN A4,1
 (54) SKOCZ W1
- (55) KONIEC
- (56) /*LICZENIE TERMU*/
- (57) TERM: PROC
 (58) WYKONAJ ARGUMENT
- (59) OPERATOR: NAZWA MN: '***', DZ: '/*'
- (60) T1: SKONA A1, OPERATOR
 (61) WROC
- (62) MN: WYKONAJ ARGUMENT
 (63) PRZESLIJ A3,B4,1
 (64) MNOZ A3,B4
 (65) USUN A4,1
 (66) SKOCZ T1
- (67) DZ: WYKONAJ ARGUMENT
 (68) PRZESLIJ A3,B4,1
 (69) DZIEL A3,B4
 (70) USUN A4,1
 (71) USUN A3,1
 (72) SKOCZ T1
- (73) KONIEC
- (74) /*LICZENIE ARGUMENTU*/
- (75) ARGUMENT: PROC
 (76) RW B1,D
 (77) SKOPL CALKOWITA
 (78) RW A1, '('
 (79) SKOMI BLAD
 (80) WYKONAJ FORMULA
 (81) RW A1, ')'
 (82) SKOMI BLAD
 (83) WROC

- (84) CALKOWITA: LICZBA A1
 (85) PRZESLIJ A1, B3, 1
 (86) WROC
- (87) KONIEC
- (88) /*SYGNALIZACJA BLEDU*/
- (89) BLAD: PROC
 (90) USUN A3
 (91) UMIESC '*LW FORMULE : *L ', Q1
 (92) PISZ A2, Q1
 (93) RW A1, '*L'
 (94) SKOPL BL1
 (95) UMIESC '*LBLAD OD MIEJSCA : ', Q1
 (96) PISZ A1, Q1
 (97) UMIESC '*L', Q1
 (98) SKOCZ WCZYT
- (99) BL1: UMIESC '*LZLY KONIEC *L', Q1
 (100) SKOCZ WCZYT
- (101) KONIEC
- (102) /*KONIEC DANYCH*/
- (103) KODAN: STOP
- (104) KONS

KONP ARYT

O b j a ś n i e n i e p r o g r a m u

- (3) - Wypisanie tytułu wyników.
- (5) - (11) Usuwanie początkowych spacji, znaków powrotu karetki i nowej linii.
- (12) - (21) Ta część programu wczytuje kolejny wiersz do wyrażenia E1 w ten sposób, że liczby całkowite bez znaku i znaki klasy R tworzą oddzielne składniki. Wewnętrzne spacje i znaki powrotu karetki są usuwane. Np. wiersz

$$12 \sqcup + \sqcup (43 - 5 * 5)$$

zostanie wczytany następująco:

$$E1 : ^{-}12^{-} + ^{-} (^{-}43^{-} - ^{-}5^{-} * ^{-}5^{-})$$

- (25) - (26) Badanie czy nastąpił koniec danych i ewentualne przejście do zakończenia.
- (28) Zapamiętanie kopii formuły w wyrażeniu E2.
- (29) Umieszczenie specjalnego składnika dla łatwiejszego określenia końca formuły.
- (30) - (38) Składnia naszej formuły zapisana formalnie jest następująca:
- $$\langle \text{formuła arytmetyczna} \rangle ::= \langle \text{term} \rangle$$
- $$[\{ + | - \} \langle \text{term} \rangle] ..$$
- $$\langle \text{term} \rangle ::= \langle \text{argument} \rangle [\{ * | / \}$$
- $$\langle \text{argument} \rangle] ..$$
- $$\langle \text{argument} \rangle ::= \langle \text{całkowita nie-} \\ \text{znakowana} \rangle |$$
- $$(\langle \text{formuła arytmetyczna} \rangle)$$

Poszczególnym pojęciom metajęzycznym odpowiadają podprogramy:

FORMUŁA, TERM i ARGUMENT.

Odwołanie do podprogramu F O R M U Ł A powoduje usunięcie tych wszystkich składników, które tworzą formułę na początku wyrażenia E1. Wartość tej formuły przesyłana jest przez podprogram na początek E3. Jeżeli wiersz danych był poprawny, to w E1 powinien pozostać jedynie składnik umieszczony tam rozkazem (29). Jeśli

tak jest, następuje drukowanie wyniku. W przeciwnym razie program przechodzi do sygnalizacji błędu.

- (40) - (55) W podprogramie FORMUŁA wykonuje się najpierw podprogram TERM. Usuwa on początkowy term z E1 przesyłając przy tym wartość termu na początek E3. Z kolei bada się w pętli, czy na początku E1 znajduje się składnik $+$ lub $-$. Jeśli tak, to ponownie wykonuje się podprogram TERM, a dwie liczby z początku E3 zastępuje się odpowiednio sumą lub różnicą tych liczb. Jeśli w kolejnym przejściu pętli nie ma owych składników, następuje powrót z podprogramu.
- (57) - (73) Podprogram TERM działa analogicznie jak podprogram FORMUŁA.
- (75) - (87) Gdy wywoływany jest podprogram ARGUMENT na początku E1 powinien znajdować się tekstowy składnik złożony z cyfr lub z jednego znaku - nawiasu otwierającego. W pierwszym przypadku składnik ten jest zamieniany na liczbę i przesyłany na początek E3. W drugim przypadku korzysta się z podprogramu FORMUŁA usuwając przy tym odpowiednie nawiasy. W razie wystąpienia niezgodności ze składnią sygnalizowany jest błąd.

W y j ś c i e. Na wyjściu program wypisuje 10 wygenerowanych liczb pseudolosowych, przy czym każda liczba wypisywana jest w oddzielnym wierszu.

P r o g r a m

- (1) SEKCJA DRUK
- (2) ROBOCZE 1
- (3) UMIESC 0, B2
- (4) /*REZERWOWANIE SKLADNIKA NA LICZBE LOSOWA*/
- (5) UMIESC 'XXXX', B1
- (6) A: IDZ LOS
- (7) UMIESC '*L', Q1
- (8) SLOWO B1
- (9) PISZ A1, Q1, 1
- (10) DODAJ A2, 1
- (11) RW B2, 10
- (12) SKOMI A
- (13) STOP
- (14) KONS

- (15) SEST LOS
- (16) UMM S/*GENEROWANIE LICZBY LOSOWEJ*/
- (17) MNM S1
- (18) PMM S
- (19) MNM ZAKRES
- (20) /*PRZESLANIE LICZBY LOSOWEJ DO 1-GO
- (21) SKLADNIKA WYRAZENIA*/
- (22) UMB 7607+2
- (23) NZB
- (24) PZA 1+
- (25) UMA 7428*
- (26) DOA +5
- (27) PZA 7428*
- (28) SKO 7429*
- (29) S:=14631461
- (30) S1:=14631461
- (31) ZAKRES:+1000
- (32) KONS
- (33) KONP DRUK

O b j a ś n i e n i a p r o g r a m u

- (2) Rezerwacja pamięci roboczej.
W przypadku, gdy część programu stanowią sekcje stałe w SAS-ie,

brak tej deklaracji może spowodować zniszczenie tych sekcji.

- (3) Ustawienie licznika liczb.
- (5) Rezerwacja składnika w wyrażeniu E1 na liczbę losową.
- (6) Przejście do sekcji w SAS-ie w celu wygenerowania liczby losowej.
- (7) - (12) Wydrukowanie liczby i sprawdzenie czy wydrukowano 10 liczb.
- (16) - (19) Generowanie liczby losowej.
- (22) - (24) Przesłanie wygenerowanej liczby losowej do pierwszego składnika wyrażenia E1.
- (25) - (28) Powrót z sekcji SAS do rozkazu oznaczonego numerem (7) w sekcji EOL.

W y n i k i

520
255
636
199
362
803
332
768
310
917

DODATEK A
ZBIÓR ZNAKÓW DLA DANYCH

Uporządkowanie znaków dla danych programu w języku EOL i tekstów zawartych w programie jest zgodne z Normą Zakładową Kodu Wewnętrznego /KW-ZAM/.

Poniższa tablica zawiera pełny zbiór znaków dla danych.

spacja	L
.	M
(N
+	O
*	P
)	Q
;	R
-	S
/	T
,	U
:	V
' /apostrof/	W
=	X
[Y
]	Z

<	S1
>	0
?	1
ESC	2
SO	3
A	4
B	5
C	6
D	7
E	8
F	9
G	LF
H	◇
I	FF
J	CS
K	CR
	DEL

Sposób przyporządkowania znakom kodu wewnętrznego znaków kodów zewnętrznych zawarty jest w opracowaniu "System operacyjny cz. I, Maszyna Użytkowa Załącznik 1".

Znaki nietypowe kodu wewnętrznego mają w tekstach następującą reprezentację:

Znak kodu wewnętrznego	Reprezentacja w tekście
[* U
]	* V
<	* W
>	* X
?	* Y
ESC	* Z
SO	* M
SJ	* D

LF	* L
FF	* E
CS	* F
CR	* G
DEL	* S

W przypadku, gdy kod zewnętrzny posiada jednoznakową reprezentację nietypowego znaku kodu wewnętrznego, można jej użyć zamiast wymienionej wyżej reprezentacji dwuznakowej. Np.: w kodzie M-2 zamiast "*U" można napisać "[", tekst "*U" będzie w tym przypadku równoważny tekstowi "[".

DODATEK B

UŻYCIE W PROGRAMIE SEKCJI W MSAS-ie

W tym samym programie mogą występować zarówno sekcje w EOL-u jak i w MSAS-ie. Komunikacja między sekcjami napisanymi w różnych językach możliwa jest zasadniczo tylko poprzez odpowiednie rozkazy przejść między sekcjami. Dalej przedstawiona jest pewna inna możliwość.

Poniżej przedstawiony jest schemat pamięci wewnętrznej w czasie wykonywania programu.

0	ROBOCZE SYSTEMU OPERACYJNEGO
	EGZEKUTOR /program realizujący przejścia międzysekcyjne/
	SEKCJA W EOL-u LUB SEKCJA WYMIENNA W MSAS-ie
	PAMIĘĆ ROBOCZA /tylko dla sekcji EOL-u/
	SEKCJA STAŁA W MSAS-ie
	SEKCJA STAŁA W MSAS-ie
	INTERPRETER /program realizujący operacje EOL-u/

Z powyższego schematu wynika, że w przypadku współdziałania sekcji w EOL-u z sekcjami stałymi należy tak określać wielkość pamięci roboczej, aby ta nie nakładała się na sekcje stałe. Wielkość pamięci roboczej jest określana za pomocą deklaracji ROBOCZE /WORKSPACE/.

PPAO oznacza pojemność pamięci operacyjnej danego zestawu.

Jeśli do sekcji stałej w MSAS-ie dokonano przejścia z sekcji w EOL-u za pomocą rozkazu IDŹ /GOSEC/, to można powrócić do wykonania następnego rozkazu stosując sekwencję rozkazów:

UMA PPAO - 252 *

DOA +5

PZA PPAO - 252 *

SKO PPAO - 251 *

Takie przejście nie powoduje zerowania wyrażeń ani SAR. Interpreter EOL-u wykorzystuje numery przesłań 28, 29, 30, 31. Interpreter definiuje swe przesłania przy każdym przejściu z MSAS-u do EOL-u za pomocą makrorozkazu IDŹ DO SEKCJI.

Natomiast przy przejściu opisanym powyżej, przesłania nie są definiowane i wymienionych wyżej numerów nie wolno używać. Wymiana informacji między sekcjami różnojęzycznymi odbywa się zasadniczo poprzez wyrażenia.

Budowa wyrażeń

Wyrażenie zapisane jest w pamięci roboczej jako ciąg tzw. ogniów. Ogniwo jest parą kolejnych słów, lecz same ogniwa nie muszą być w pamięci rozmieszczone kolejno.

Pierwsze słowo ogniwa zawiera adres następnego ogniwa, drugie zaś informację. Ostatnie ogniwo zamiast adresu zawiera liczbę -0. Składnik zapisywany jest w całkowitej ilości ogniw. Ostatnie ogniwo składnika zawiera 1 w pozycji znaku słowa zawierającego adres. W słowie informacyjnym mieści się do czterech znaków kodu wewnętrznego lub liczba binarna. Znaki zapisywane są od lewej strony słowa poczynając. Jeśli znaków jest mniej niż cztery, słowo dopełnia się znakami pustymi.

Adres pierwszego ogniwa i-tego wyrażenia mieści się w PPA0-73+2i-tym słowie pamięci. W następnym słowie mieści się adres ostatniego ogniwa tego wyrażenia. Oba adresy mają 1 w pozycji znaku. Jeśli wyrażenie jest puste oba adresy mają wartość -0.

P r z y k ł a d

Niech

E2: -ABCDEF °5 -OKO- +12

Wyrażenie to może być zapisane w pamięci maszyny następująco:

adres	zawartość	typ zawartości
1200	1350	binarna
1201	ABCD	znakowa
1350	-1402	binarna
1351	EFpp	znakowa
1402	-1500	binarna
1403	+5	binarna
1500	-1450	binarna
1501	OKOp	znakowa

1450 -0 binarna

1451 +12p znakowa

p- oznacza znak "pusty".

Podany niżej fragment programu ilustruje przekazywanie informacji między sekcjami różnojęzycznymi dla PPAO równej 7680.

SEKCJA TR

- - - - -

- - - - -

/* W PONIŻSZEJ PROCEDURZE PIERWSZY SKŁADNIK WYRAŻENIA E1 ZAWIERAJĄCY LICZBĘ ZMIENNOPRZECINKOWĄ W POSTACI DZIESIĘTNEJ ZOSTAJE ZASTĄPIONY DWUOGNIWOWYM SKŁADNIKIEM ZAWIERAJĄCYM TĘ LICZBĘ W POSTACI BINARNEJ. PROCEDURA WYKONUJE PRZEJŚCIE DO M-SAS-U*/.

FLT:PROC

UMIEŚĆ 'XXXXX', A2 /*REZERWACJA SKŁADNIKA NA WYNIK*/

IDŹ ZM

USUŃ A1,1

PRZEŚLIJ A2, B1

WRÓĆ

KONIEC

- - - - -

- - - - -

KONS

SEST ZM

/*SEKCJA TA TŁUMACZY LICZBĘ ZMIENNOPRZECINKOWĄ MIESZCZĄCĄ SIĘ W PIERWSZYM SKŁADNIKU WYRAŻENIA E1 NA POSTAĆ BINARNĄ I UMIESZCZA W PIERWSZYM SKŁADNIKU WYRAŻENIA E2 WYMAZUJĄC POPRZEDNIA ZAWARTOŚĆ*/.

- - - - -

- - - - -

UMA 7428*

DOA +5

PZA 7428*

SKO 7429*

KONS

DODATEK C

PRZYPISYWANIE URZĄDZEŃ ZMIENNIM WEJŚCIOWYM
/WYJŚCIOWYM/

Zmienna $I \langle n \rangle / Q \langle n \rangle /$ przyporządkowana jest $n-1$ -emu symbolicznemu wejściu /wyjściu/ systemu operacyjnego.

Posługując się zmiennymi $I \langle n \rangle / Q \langle n \rangle$ dla $n > 1$ należy przedtem dołączyć odpowiedni podprogram i dekodery. Dołączenia dokonuje się w MSAS-ie. Określając obszar na podprogram i dekodery należy uwzględnić rozmieszczenie programu głównego, pamięci roboczej i interpretera.

DODATEK D

 SYGNALIZACJA BŁĘDÓW
 I PRZYCZYNY ZAKOŃCZENIA PROGRAMU

1. W czasie translacji programu w języku EOL sygnalizowane są następujące błędy:

Numer błędu	T r e ś ć
00	Brak nazwy sekcji
01	Za dużo argumentów
02	Niewłaściwy argument
03	Brak testu
04	Niewłaściwy test lub niewłaściwy argument
05	Niepoprawna struktura blokowa
06	Podwójna etykieta
07	Podwójne wystąpienie deklaracji START
08	Brak nazwy procedury lub nazwy zwrotnicy
09	Łączna długość programu i zadeklarowanej pamięci roboczej przekracza pojemność PAO
10	Niepoprawny rozkaz

Numer błędu	T r e ś ć
11	Przekroczono pojemność tablicy liczb lub napisów
12	Niezareklamowana etykieta
13	Niezareklamowana zwrotnica
14	Użyto niedozwolonego znaku
15	Niedozwolony znak w tekście
16	Liczba, tekst lub identyfikator przekracza 60 znaków
17	Niepoprawna nazwa lub parametr makrodefinicji
18	Identyczne nazwy makrodefinicji
19	Użyto niezareklamowanego parametru formalnego
20	Odwołanie do nieistniejącej makrodefinicji
21	Niegodna ilość parametrów i argumentów makrodefinicji
22	Przekroczono listę odwołań do makrodefinicji
23	Łączna długość przedrostka i argumentu przekracza 60 znaków
24	Niepoprawne zakończenie listy makrodefinicji
25	Niepoprawne zakończenie programu
26	Niepoprawne zakończenie odwołania wewnętrznego

Informacje o błędach występujących w danej sekcji poprzedzone są tekstem SEKCJA, po którym wypisywana jest nazwa sekcji.

Poszczególne rodzaje błędów sygnalizowane są następująco:

- a/ Błąd 00 sygnalizowany jest tekstem
BLAD 00.
- b/ Błędy 12 i 13 sygnalizowane są za pomocą tekstu BLAD, po którym następuje numer błędu, a po nim niezadeklarowana etykieta /12/ lub nazwa niezadeklarowanej zwrotnicy /13/, np.:

BLAD 12 SORT

oznacza, że w sekcji programu nie zadeklarowano etykiety SORT.

- c/ Pozostałe błędy są sygnalizowane w następujący sposób: Wypisywane jest słowo BLAD, po którym następuje numer błędu a po nim etykieta i liczony względem tej etykiety numer wiersza, w którym wystąpił błąd. Jeżeli błąd wystąpił przed pojawieniem się pierwszej deklaracji etykiety, to zamiast etykiety wypisywane są trzy gwiazdki, a numer wiersza liczony jest względem pierwszego wiersza sekcji nie wliczając makrodefinicji. Np.: BLAD 10 RELACJA 04 oznacza, że w czwartym wierszu względem etykiety RELACJA wystąpił niepoprawny rozkaz^{*}. Błędy w makrodefinicji sygnalizowane są podobnie, z tym że zamiast etykiety wypisywana jest nazwa makrodefinicji, a numer wiersza jest liczony względem wiersza zawierającego tę nazwę.

W przypadku, gdy sekcja programu została napisana w wersji angielskiej zamiast SEKCJA wypisywane jest słowo SECTION, a zamiast słowa BLAD - słowo ERROR.

^{*}/ Przy lokalizacji błędu należy uwzględnić, że komentarze nie są brane pod uwagę, a zamiast odwołań do makrodefinicji wstawia się do programu treść makrodefinicji.

2. Przyczyny zakończenia programu

W programach EOL-u są wykorzystywane pewne numery przyczyn zakończenia dozwolone dla programów użytkowych. Znaczenie tych numerów jest następujące:

- 99 - wykonanie operacji STOP.
- 100 - przepełnienie pamięci roboczej /w pamięci tej zapisywane są wyrażenia i SAR/.
- 101 - próba wykonania operacji plikowej, jeśli kiedykolwiek przedtem w danej sekcji zapisano więcej niż $n-1$ bloków pamięci roboczej, gdzie n - ilość zadeklarowanych bloków.
- 102 - przepełnienie pamięci bębnowej /w pamięci bębnowej zapisywane są pliki/.

DODATEK E

LISTA SŁÓW KLUCZOWYCH

COFNIJ	RESET	RZ	ER
CZYTAJ	READ	SCHOWAJ	SAVE
DODAJ	ADD	SEKCJA	SECTION
DZIEL	DIV	SKOCZ	GOTO
IDŹ	GOSEC	SKOIN	GGIN
INDEKS	INDEX	SKOMI	GOMI
KONIEC	END	SKONA	GONA
KONP	KONP	SKOPL	GOPL
KONS	ENDS	SKREŚL	DELETE
KOPIUJ	COPY	SŁOWO	WORD
LICZBA	NUMBER	START	START
MN	LT	STOP	STOP
MNÓŻ	MULT	SZL	SEC
NS	FL	SZUKAJ	SEARCH
NAZWA	NAME	UMIEŚĆ	SET
ODEJMIJ	SUB	USTAW	RESTORE
OML	SHIC	USUŃ	CLEAR
OMIŃ	SHIFT	WK	GT
PAKUJ	PACK	WRÓC	RETURN
PISZ	WRITE	WSTAW	PUT
POBIERZ	GET	WYIN	CAIN
POLE	SPACE	WYKONAJ	CALL
PP	PR	WYMI	CAMI

PROC	PROC	WYNA	CANA
PRZEŚLIJ	MOVE	WYPL	CAPL
ROBOCZE	WORKSPACE	ZAMIEŃ	EXCHANGE
ROZBIJ	SPLIT	ZBIJ	COMPRESS
RW	EQ	ZNAJDŹ	FIND

Przy wprowadzeniu do maszyny program w języku EOL dekodowany jest na postać wewnętrzną zgodnie z regułami użytego dekodera. Tak więc, translator EOL-u przetwarza program przedstawiony w kodzie wewnętrznym. Podczas translacji programu pomijany jest znak: r5, o ile nie występuje wewnątrz tekstów.

Jak wynika z tabeli KW-ZAM podanej w Dodatku A znaki: Ć, Ń, 0, Ś, Ż, Ź w kodzie wewnętrznym nie występują. W opisie użyto ich wyłącznie dla przejrzystości zamiast znaków: C, N, O, S, Z, Z.

DODATEK F
SKOROWIDZ TERMINÓW

A	4.2.1.1.
A	4.2.1.2.
A	4.2.1.3.
ADD	5.6.
ADRES ROZKAZU	3.6.
ADRES ZAPISU	3.3.3.
ARGUMENT	4.2.
ARGUMENT	7.
ARGUMENT WEWNĘTRZNY	7.
B	4.2.1.1.
B	4.2.1.2.
B	4.2.1.3.
B	4.2.3.
C	4.2.1.1.
C	4.2.1.2.
CAIN	5.5.3.
CALL	5.5.1.
CAŁKOWITA	3.3.2.
CAŁKOWITA	4.2.5.
CAMI	5.5.2.
CANA	5.5.3.

CAPL	5.5.2.
CIĄG ZNAKÓW	3.1.
CLEAR	5.3.
COFNIJ	5.10.1.
COMPRESS	5.7.3.
COPY	5.1.4.
CYFRA	2.1.2.
CZYTAJ	5.1.2.
CZYTANIE	4.2.11.
D	4.2.1.1.
D	4.2.1.2.
D	4.2.3.
DEKLARACJA	5.11.
DELETE	5.3.
DIV	5.6.
DODAJ	5.6.
DOPISYWANIE	4.2.1.2.
DZIEL	5.6.
E	4.2.1.3.
ELEMENT MAKRODEFINICJI	7.
ELEMENT SEKCJI MODYFIKOWANEJ	7.
END	6.1.
ENDS	6.1.
EQ	5.4.1.
EQ	5.4.2.
ER	5.4.3.
ETYKIETA	4.2.2.
EXCHANGE	5.1.9.
FIND	5.9.1.
FL	5.4.2.
GET	5.1.6.
GOIN	5.5.3.
GOMI	5.5.2.
GONA	5.5.3.

GOPL	5.5.2.
GOSEC	5.10.4.
GOTO	5.5.1.
GT	5.4.1.
I	4.2.1.1.
I	4.2.1.3.
IDENTYFIKATOR	4.2.2.
IDŹ	5.10.4.
ILOŚĆ	5.1.1.
INDEKS	4.2.1.
INDEKS	5.11.1.
K	4.2.1.4.
KLASA	4.2.3.
KOMENTARZ	5.12.
KONIEC	6.1.
KONP	6.1.
KONS	6.1.
KOPIUJ	5.1.4.
L	4.2.3.
LICZBA	3.3.2.
LICZBA	5.7.2.
LITERA	2.1.1.
LITERA ARGUMENTU	4.2.1.
LITERA INDEKSOWANA	4.2.1.
LISTA MAKRODEFINICJI	7.
LISTA PARAMETRÓW	7.
LISTA ZWROTNICY INDEKSOWEJ	5.11.1.
LISTA ZWROTNICY NAZWOWEJ	5.11.1.
LT	5.4.1.
MAKRODEFINICJA	7.
MN	5.4.1.
MNÓŻ	5.6.
MOVE	5.1.1.
MULT	5.6.

N	4.2.1.4.
NAZWA	5.11.1.
NAZWA MAKRODEFINICJI	7.
NAZWA PROCEDURY	6.1.
NAZWA SEKCJI	5.10.4.
NAZWA SEKCJI	6.1.
NAZWA ZWROTNICY	5.11.1.
NOTACJA WARTOŚCI ZMIENNEJ	1.5.
NOWA LINIA	4.2.4.
NS	5.4.2.
NUMBER	5.7.2.
ODEJMIJ	5.6.
ODWOŁANIE	7.
ODWOŁANIE DO MAKRODEFINICJI	7.1.
ODWOŁANIE WEWNĘTRZNE	7.
OML	5.9.3.
OMIŃ	5.8.2.
OMIŃ I LICZ	5.9.3.
OPERATOR	4.1.
P	4.2.1.4.
PACK	5.10.2.
PAKUJ	5.10.2.
PARAMETR	7.
PISZ	5.1.3.
PLIK	3.4.
POBIERZ	5.1.6.
POLE	4.2.5.
PP	5.4.2.
PR	5.4.2.
PROC	6.1.
PROCEDURA	6.1.
PROGRAM	6.1.
PRZEKSZTAŁCENIE	5.7.
PRZEKSZTAŁCENIE NA LICZBĘ	5.7.2.

PRZEKSZTAŁCENIE NA SŁOWO	5.7.1.
PRZESŁANIE	5.1.
PRZEŚLIJ	5.1.1.
PUT	5.1.5.
Q	4.2.1.2.
R	4.2.3.
READ	5.1.2.
RELACJA	5.4.
RELACJA ARYTMETYCZNA	5.4.1.
RELACJA ARYTMETYCZNA RÓWNOŚCI	5.4.1.
RELACJA ARYTMETYCZNA UPORZĄDKOWANIA	5.4.1.
RELACJA TESTOWA	5.4.2.
RELACJA TESTOWA RÓWNOŚCI	5.4.2.
RELACJA TESTOWA UPORZĄDKOWANIA	5.4.2.
RELACJA ZAPISU	5.4.3.
RESET	5.10.1.
RESTORE	5.1.8.
RETURN	5.5.4.
ROBOCZE	5.11.2.
ROZBIJ	5.7.4.
ROZBIJANIE	5.7.4.
ROZKAZ	4.1.
ROZKAZ	5.
ROZKAZ ARYTMETYCZNY	5.6.
ROZKAZ STERUJĄCY	5.5.
RW	5.4.1.
RW	5.4.2.
RZ	5.4.3.
SAVE	5.1.7.
SCHOWAJ	5.1.7.
SEARCH	5.8.1.
SEC	5.9.2.
SECTION	6.1.
SECTION	7.
SEKCJA	6.1.

SEKCJA	7.
SEKCJA MODYFIKOWANA	7.
SEKCJA M-SAS	6.1.
SEKCJA M-SAS	B-1
SEKCJA ŹRÓDŁOWA	7.
SET	5.2.
SHIC	5.9.3.
SHIFT	5.8.2.
SKŁADNIA	1.3.
SKŁADNIK	3.3.
SKOCZ	5.5.1.
SKOIN	5.5.3.
SKOK	5.5.1.
SKOK POWROTNY	5.5.4.
SKOK WARUNKOWY	5.5.2.
SKOK ZWROTNICY	5.5.3.
SKOMI	5.5.2.
SKONA	5.5.3.
SKOPL	5.5.2.
SKREŚL	5.3.
SŁOWO	3.3.1.
SŁOWO	5.7.1.
SPACE	4.2.5.
SPLIT	5.7.4.
START	5.11.3.
START PROGRAMU	5.11.3.
STOP	5.10.3.
STOS ADRESÓW ROZKAZÓW	3.6.
SZL	5.9.2.
SZUKAJ	5.8.1.
SZUKAJ I LICZ	5.9.2.
SZUKANIE	5.8.
SUB	5.6.
SYMBOLE PODSTAWOWE	2.
T	4.2.1.3.
TEKST	4.2.4.

TEST KOPIOWANIA	5.1.4.
TEST PRZESŁANIA	5.1.1.
TREŚĆ MAKRODEFINICJI	7.
TRESC SEKCJI MODYFIKOWANEJ	7.
UMIESZCZENIE	5.2.
UMIEŚĆ	5.2.
UMIEŚĆ NA WYJŚCIU	5.2.
UMIEŚĆ W PLIKU	5.2.
UMIEŚĆ W WYRAŻENIU	5.2.
USTAW	5.1.8.
USUNIĘCIE	5.3.
USUŃ	5.3.
USUŃ Z PLIKU	5.3.
USUŃ Z WEJŚCIA	5.3.
USUŃ Z WYRAŻENIA	5.3.
WARTOŚCI POCZĄTKOWE	3.7.
WARTOŚĆ LOGICZNA	3.5.
WARTOŚĆ PLIKU	3.4.
WARTOŚĆ STOSU ADRESÓW	3.6.
WARTOŚĆ WEJŚCIA	3.1.
WARTOŚĆ WYJŚCIA	3.2.
WARTOŚĆ WYRAŻENIA	3.3.
WEJŚCIE	3.1.
WERSJE JĘZYKA EOL	1.4.
WK	5.4.1.
WORD	5.7.1.
WORKSPACE	5.11.2.
WRITE	5.1.3.
WRÓĆ	5.5.4.
WSTAW	5.1.5.
WYIN	5.5.3.
WYJŚCIE	3.2.
WYKONAJ	5.5.1.
WYKORZYSTANIE	4.2.1.3.
WYMI	5.5.2.

WYNA	5.5.2.
WYPL	5.5.2.
WYRAŻENIE	B-2
WYRAŻENIE	3.3.
Y	4.2.1.2.
Z	4.2.1.2.
ZAMIEŃ	5.1.9.
ZAPIS	3.4.
ZBIJ	5.7.3.
ZBIJANIE	5.7.3.
ZBIÓR ZNAKÓW DLA DANYCH	A-1
ZDANIE	6.1.
ZDANIE ZĘWNETRZNE	6.1.
ZLICZANIE	5.9.
ZMIENNA	3.
ZMIENNA LOGICZNA	3.5.
ZNACZNIK	1.5.
ZNAJDŹ	5.9.1.
ZNAK	2.1.
ZNAK SPECJALNY	2.1.3.
ZWROTNICA	5.11.1.
ZWROTNICA INDEKSOWA	5.11.1.
ZWROTNICA NAZWOWA	5.11.1.

SYMBOLE

*	4.2.4.
**	4.2.4.
*L	4.2.4.
*'	4.2.4.
,	4.2.4.
{	1.3.
[1.3.
]	1.3.
]	1.3.
λ	2.1.
┌	2.1.

