

ПРОВЕРКА НА ЛОГИЧЕСКАТА СТРУКТУРА НА ПРОГРАМИ ЗА АСМ „МИНСК-2“

Петър Бърнев и Александър Ахегукян

При програмирането на машинен или машинно-ориентиран език често се допускат грешки. Тези грешки могат да се разделят на два типа: грешки при програмната реализация на отделните блокове от блок-схемата на алгоритъма и грешка при реализацията на връзките между отделните блокове.

Предмет на настоящата работа е създаването на средства за откриване на грешките от втория тип.

За да се улесни проверката на логическата структура на дадена програма, могат да се използват следните програми:

- а) програма „БС“ за създаване на блок-схема на дадената програма;
- б) програма „Проследяване“, която да анализира последователността на изпълнение на отделните блокове от дадената програма.

Програмата „БС“ позволява да се анализира общата структура на проверяваната програма, като се установят различните блокове и връзката между тях. Тази програма не позволява обаче да се установи какво ще бъде конкретното свързване на блоковете при зададени начални данни, т. е. кои от възможните връзки между блоковете ще бъдат използвани и в какъв ред. Програмата „БС“ не взима под внимание модификациите на адресите в командите за предаване на управлението.

Програмата „Проследяване“ предизвиква изпълнение на анализиранията програма със съответни начални данни, при което се следи и отпечатва информация за измененията в естествения ред на изпълнение на командите. Това позволява да се проследи последователността на изпълнение на отделните блокове в зависимост от началните данни. Трябва да се отбележи, че програмата „Проследяване“ забавя неколkokратно изпълнението на анализиранията програма, поради което трябва да се използва само при необходимост.

В първите две части на настоящата работа се разглеждат съответно програмите „БС“ и „Проследяване“, предназначени за изследване на програми, написани на машинния език на „Минск-2“ [1] или на езика за символично програмиране [2] и включени в системата МИД-2 [3]. В третата част се привеждат някои примери.

Авторите изказват благодарност за направените забележки и оказаната помощ от страна на сътрудниците на Изчислителния център на Математическия институт, където бе извършена разглежданата работа.

1. ПРОГРАМА „БС“

Програмата „БС“ анализира проверяваната програма преди нейното изпълнение, разделя я на блокове и отпечатва съответна блок-схема.

Ще различаваме няколко типа блокове:

аритметичен блок, състоящ се от група последователни команди, не-съдържащи команди за предаване на управлението;

Т а б л и ц а 1

№	Тип на блока	Кодировка при печат	
		БПМ-20	СТА-2М
1	Аритметичен	$+ a 0 n$	$ \begin{array}{c} + + + + + \\ a - + \quad + - n \\ + + + + + \end{array} $
2	Аритметичен с последна команда БПУ	$+ a n 0$	$ \begin{array}{c} + + + + + \\ a - + \quad + \\ + + + + + \\ n \end{array} $
3	Преход към подпрограма	$- a 0 A$	$ \begin{array}{c} + + + + + \\ a - + \quad A + \\ + + + + + \end{array} $
4	Преход към подпрограма със следваща команда БПУ	$- a n A$	$ \begin{array}{c} + + + + + \\ a - + \quad + \\ + + + + + \\ n \end{array} $
5	Преход към СП	$- a 0 N$	$ \begin{array}{c} + + + + + \\ a - + \quad N + \\ + + + + + \end{array} $
6	Преход към СП със следваща команда БПУ	$- a n N$	$ \begin{array}{c} + + + + + \\ a - + \quad N + \\ + + + + + \\ n \end{array} $
7	Логически	$ \begin{array}{c} + 0 0 0 \\ + a n_1 n_2 \end{array} $	$ \begin{array}{c} + + \\ + \quad + \\ a - + \quad + - n_2 \\ + \quad + \\ + + \\ n_1 \end{array} $
8	Край на цикъл	$ \begin{array}{c} + 0 0 i \\ + a n_1 n_2 \end{array} $	$ \begin{array}{c} + + \\ + \quad + \\ a - + \quad i \quad + - n_1 \\ + \quad + \\ + + \\ n_2 \end{array} $

преход към стандартната подпрограма (СП), състоящ се от команда за обръщение към СП и евентуално група команди с параметри съгласно изискванията на системата МИКС за ползване на библиотеката СП [4];

преход към подпрограма, състоящ се от една единствена команда с код на операцията (КОП) — 31;

логически блок, състоящ се от една единствена команда за условно предаване на управлението (КОП = —32, —33, —34 или —35).

край на цикъл, състоящ се от командата с КОП = —20.

Командите за безусловно предаване на управлението (БПУ) се обединяват с намиращия се непосредствено пред тях блок. Ако след аритметичен блок, блок за переход към СП или блок за переход към подпрограма следва оператор за БПУ, той се обединява със съответния блок. Поради това при кодирането на тези блокове съществуват по две разновидности. В другите два типа блокове (логически и край на цикъл) разновидности при кодирането няма, но и при тях се включва следващият оператор, ако той е за БПУ.

В останалите случаи командата за БПУ се разглежда като аритметичен блок с БПУ, като аритметичният блок има празно съдържание.

В табл. 1 се привежда кодиранката на различните типове блокове при отпечатването им на тясна хартиена лента чрез устройството БПМ-20 и на широка хартиена лента чрез устройството СТА-2М.

В таблицата с a е отбелязан началният адрес на блока, с n или n_1 и n_2 — адресите на блоковете наследници, с A — началният адрес на подпрограмата, с N — номерът на СП и с i — номерът на индексната клетка, участваща в командата за край на цикъл.

В случая, когато блокът има два наследника, с n_1 е означен наследникът, съответстващ на първия адрес на командата за предаване на управлението. При отпечатването a , n , n_1 , n_2 , A и N се състоят от по четири осмични цифри, а i от две осмични цифри. Символът 0 означава четири осмични нули.

Алгоритъмът, по който работи програмата „БС“, се състои от две части. Най-напред се създава списък от адресите, които съдържат команди за управление, и адресите, на които се предава управлението. Този списък се подрежда по големина на адресите без повторение. След това полученият списък се обработва последователно в зависимост от текста на командите, чиито адреси са указани, и се кодира по съответния начин, така че да може лесно да бъде отпечатан.

Подробно функционирането на алгоритъма може да се проследи от приложената блок-схема (фиг. 1), където са използвани следните означения:

α — начален адрес на програмата;

ω — краен адрес на програмата;

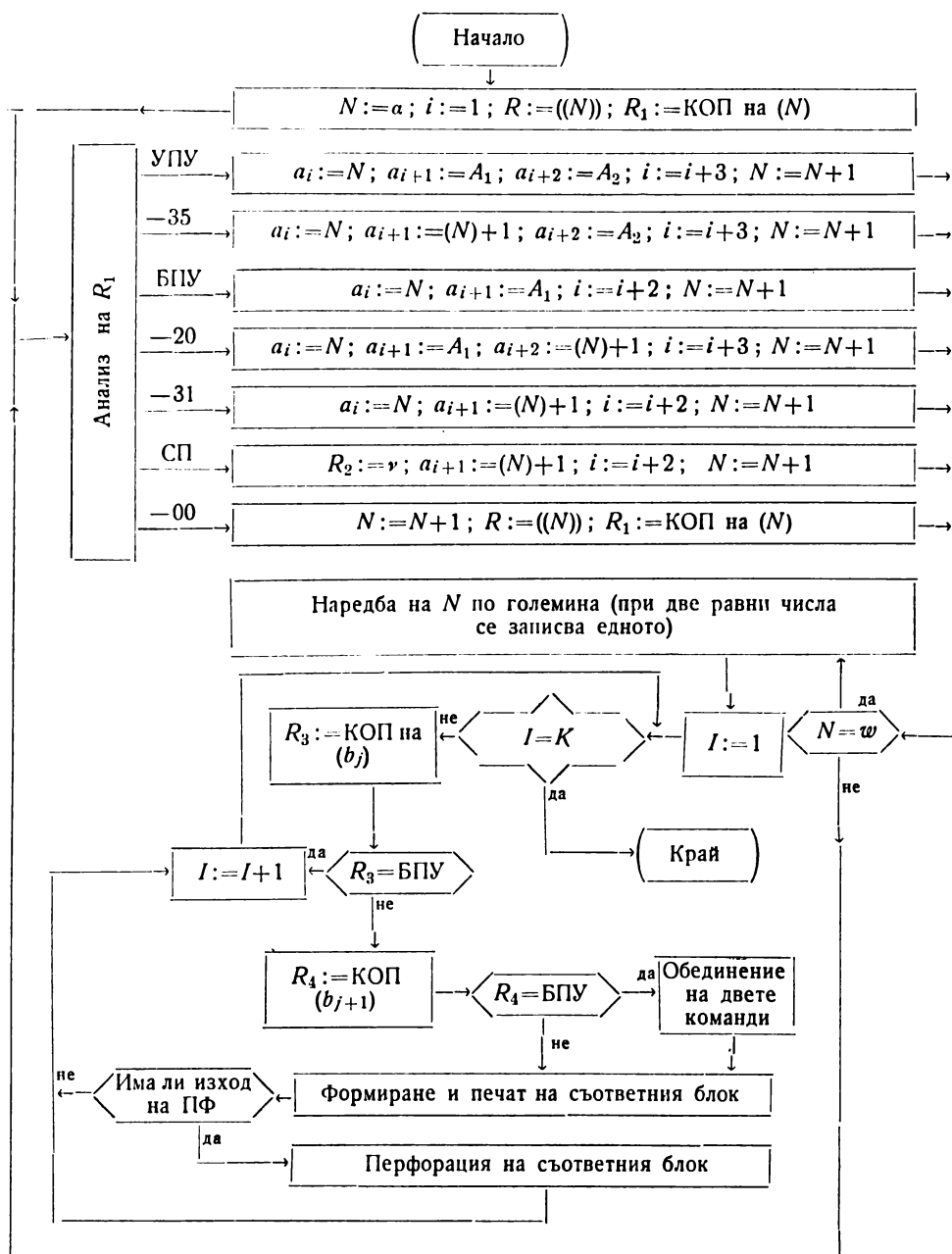
N — адрес на командата;

A_1 — първа адресна част на командата;

A_2 — втора адресна част на командата;

(N) — съдържанието на клетка с адрес N .

Програмата „БС“ заема 765₍₈₎ клетки в оперативната памет.



Фиг. 1

2. ПРОГРАМА „ПРОСЛЕДЯВАНЕ“

Програмата „Проследяване“ изпълнява чрез интерпретация анализиранията програма команда по команда. При срещане на команди с положителен код на операцията (КОП), както и на команда с отрицателен

КОП от тип α , т. е. КОП = -04, -05, -06, -07, -10, -11, -12, -13, -14, -15, -16, -36, -40, -41, -43, -44, -45, -46, -47, -50, -51, -52, -53, -60, -61, -70, -71, -72, -73, -74, -75, -76, се преминава към изпълнение на следващата команда, без да се отпечатва индикация.

При срещане на команди с отрицателен КОП от тип β , т. е. КОП = -20, -30, -31, -32, -33, -34, -35, се определя наследникът на съответната команда и евентуално се отпечатва индикация. Специално при команда с КОП = -30 се отпечатва индикация, съдържаща адреса на командата и нейния наследник само в случая, когато наследникът не е следващата команда.

При КОП = -32, -33, -34 и -35 винаги се отпечатва адресът на командата и на нейния наследник.

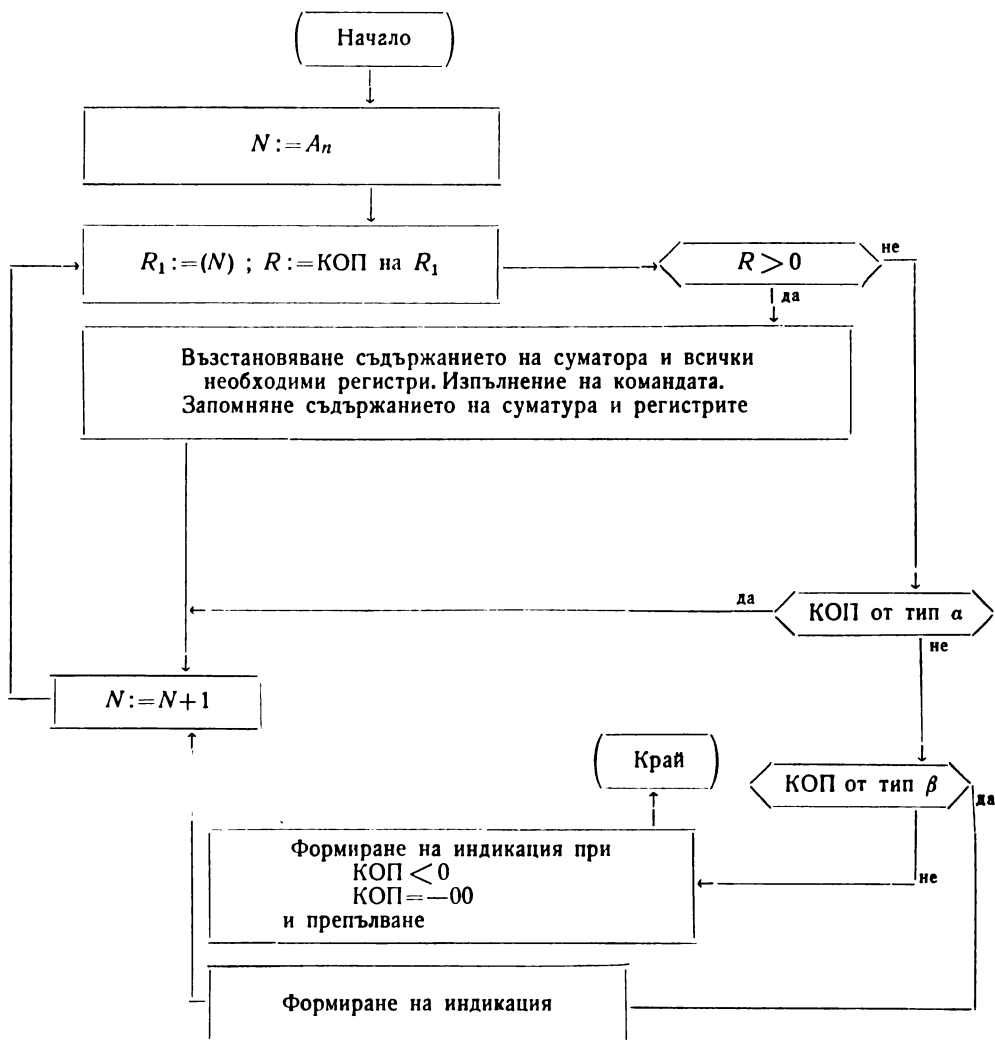
При команда с КОП = -31, която не е обръщение към СП, се постъпва по същия начин както при команда с КОП = -32, -33, -34, -35. Ако командата с КОП = -31 е обръщение към стандартна подпрограма, също се отпечатва нейният адрес и адресът на наследника ѝ, но проследяването не се извършва за командите от самата СП.

Т а б л и ц а 2

№	Пояснения	Кодировка при печат на БПМ-20
1	Начални адреси	— a
2	Команда „стоп“	+ a
3	Команда УПУ $i=1, 2$	++ a — n_i
4	•Преход към подпрограма или СП	— a — A
5	Команда БПУ	+ a — n
6	Край на цикъл при неизчерпан брояч	-i— a — n_1
7	Край на цикъл при изчерпан брояч	-i— a -+--+--+--+ n_2 —
8	Съкрати означение на цикъла, който не съдържа УПУ, БПУ и преглед към подпрограма	— n_1 -i— a — K -+--+--+--+ n_2 —

При команда с КОП = -20, съответстваща на прост цикъл, т. е. на цикъл, на който тялото не съдържа команди от тип β (допуска се сре-

щането на команди с КОП = -30, ако техният наследник е следващата команда), се отпечатва съкратена индикация, указваща началния и крайния адрес на тялото на цикъла, броя на извършените повторения и адреса на командата, която се изпълнява след завършване на цикъла.



Фиг. 2

При команда с КОП = -20, която не съответствува на прост цикъл се отпечатва индикация, съдържаща адреса на командата и нейния наследник.

Наред с изброените индикации при започване на проследяването се отпечатва началният адрес на проследяваната програма.

Проследяването завършва при срещане на команда за стоп (КОП = -00), на команда с недопустим КОП или при поява на препълване при изпълнение на съответните команди. При завършването на проследяването се отпечатва крайният адрес.

в табл. 1; k означава броя на изпълнение на тялото на цикъла.

На фиг. 2 е приведена блок-схема на програмата „Проследяване“. Програмата „Проследяване“ заема 675₍₈₎ клетки в оперативната памет.

3. ПРИМЕРИ

За илюстрация на получаваната информация чрез програмите „БС“ и „Проследяване“ се привеждат два примера.

Пример 1. Блок-схема на програмата

1100)	-10	00	1113	2000
1)	-10	00	1114	1103
2)	-10	00	1115	1105
3)	+21	00	2001	2000
4)	-32	00	1106	1105
5)	-10	00	2001	2000
6)	+10	00	1116	1103
7)	+20	00	1116	1105
1110)	+23	00	1117	0000
1)	-34	00	1103	1112
2)	-00	00	0000	0000
3)	00	00	0000	0000
4)	21	00	2001	2000
5)	-10	00	2001	2000
6)	00	00	0001	0000
7)	-10	00	2021	2000

за намиране най-голямото от 16 зададени числа, получена чрез програмата „БС“, има вида, посочени на фиг. 3.

Figure 1 shows the arrangement of 1103 and 1106 atoms in a crystal lattice. The diagram consists of three main clusters of atoms, each represented by a 3x3 square of '+' signs. The top cluster is labeled '1100' on the left and '1103' on the right. The middle cluster is labeled '1103' on the left and '1104' on the right. The bottom cluster is labeled '1105' on the left and '1106' on the right. To the right of these clusters are two more clusters: a 3x3 square of '+' signs labeled '1104' on the left and '1105' on the right, and a 3x3 square of '+' signs labeled '1106' on the left and '1103' on the right. The atoms are represented by '+' signs, and the numbers are in bold black font.

Фиг. 3

Ако се извърши проследяване на горната програма чрез програмата „Проследяване“ при следните данни:

```

2000) 00 00 0000 0000
1) 00 00 0000 0025
2) 00 00 0000 0032
3) 00 00 0000 0027
4) 00 00 0000 0021
5) 00 00 0000 0004
6) 00 00 0000 0007
7) 00 00 0000 0006
2010) 00 00 0000 0047
1) 00 00 0000 0053
2) 00 00 0000 0002
3) 00 00 0000 0057
4) 00 00 0000 0016
5) 00 00 0000 0043
6) 00 00 0000 0046
7) 00 00 0000 0047
2020) 00 00 0000 0041

```

на БПМ-20 се получава индикация, посочена на фиг. 4.

Начало		Продължение 1		Продължение 2		Продължение 3	
—	1100	++	1104	—	1106	++	1111
++	1104	—	1106	++	1111	—	1103
—	1105	++	1111	—	1103	++	1104
++	1111	—	1103	++	1104	—	1106
—	1103	++	1104	—	1106	++	1111
++	1104	—	1106	++	1111	—	1103
—	1105	++	1111	—	1103	++	1104
++	1111	—	1103	++	1104	—	1106
—	1103	++	1104	—	1106	++	1111
++	1104	—	1105	++	1111	—	1103
—	1106	++	1111	—	1103	++	1104
++	1111	—	1103	++	1104	—	1106
—	1103	++	1104	—	1106	++	1111
++	1104	—	1106	++	1111	—	1112
—	1106	++	1111	—	1103	+	1112
++	1111	—	1103	++	1104		
—	1103	++	1104	—	1106		

Фиг. 4

Пример 2. Блок-схемата на програмата

```

1002) —10 00 1022 0003
3) 71 00 1023 0003
4) 76 00 1022 0002
5) 71 00 1023 0002
6) 76 00 1022 0001
7) 24 00 1017 1017
1010) 35 01 1040 1060
1) 16 00 1017 1017
2) —20 01 1010 1021

```

```

3) -60 00 1400 1017
4) -20 02 1005 1020
5) -20 03 1003 1024
6) -00 00 0000 0000
7) 00 00 0000 0000
1020) -77 77 0000 0001
1) 00 00 0001 0003
2) 00 02 0000 0000
3) 00 00 7777 7777
4) 00 00 0003 0000

```

за намиране произведението на две дадени квадратни матрици от трети ред, получена чрез програмата „БС“, има вида, посочен на фиг. 5.

```

      + + + + +
1002 - +      + - 1003
      + + + + +

      + + + + +
1003 - +      + - 1005
      + + + + +

      + + + + +
1005 - +      + - 1010
      + + + + +

      + + + + +
1010 - +      + - 1012
      + + + + +

                        + +
                        + +
1012 - +      01      + - 1010
                        + +
                        + +
                        1013

      + + + + +
1013 - +      + - 1014
      + + + + +

                        + +
                        + +
1014 - +      02      + - 1005
                        + +
                        + +
                        1015
                        + +
1015 - +      +      + - 1003
                        + +
                        + +
                        1016

```

Фиг. 5

Ако се извърши проследяване на горната програма чрез програмата „Проследяване“ при следните данни:

```

1040) 60 00 0000 0002
1) 40 00 0000 0001
2) 40 00 0000 0001
3) 40 00 0000 0002
4) 40 00 0000 0001

```

5) 40 00 0000 0002
 6) 40 00 0000 0001
 7) 40 00 0000 0002
 1050) 60 00 0000 0003
 1060) 40 00 0000 0001
 1) 40 00 0000 0001
 2) -40 00 0000 0001
 3) 40 00 0000 0002
 4) -40 00 0000 0001
 5) 40 00 0000 0001
 6) 40 00 0000 0001
 7) 60 00 0000 0002
 1070) 40 00 0000 0001

на БПМ-20 се получава индикация, насочена на фиг. 6.

Начало	Продължение 1	Продължение 2	Продължение 3
— 1 0 0 2	— + - + - + - + - +	- 0 2 - - - 1 0 1 4	— 1 0 1 0
— 1 0 1 0	— 1 0 1 3	— 1 0 0 5	- 0 1 - - - 1 0 1 2
- 0 1 - - - 1 0 1 2	- 0 2 - - - 1 0 1 4	— 1 0 1 0	- - - - - - - - - -
- - - - - - - - - -	- + - + - + - + - +	- 0 1 - - - 1 0 1 2	— 3
— 3	— 1 0 1 5	- - - - - - - - - -	- + - + - + - + - +
- + - + - + - + - +	- 0 3 - - - 1 0 1 5	— 3	— 1 0 1 3
— 1 0 1 3	— 1 0 0 3	- + - + - + - + - +	- 0 2 - - - 1 0 1 4
- 2 0 - - - 1 0 1 4	— 1 0 1 0	— 1 0 1 3	— 1 0 0 5
— 1 0 0 5	- 0 1 - - - 1 0 1 2	- 0 2 - - - 1 0 1 4	— 1 0 1 0
— 1 0 1 0	- - - + - - - - - -	- + - + - + - + - +	- 0 1 - - - 1 0 1 2
- 0 1 - - - 1 0 1 2	— 3	— 1 0 1 5	- - - - - - - - - -
- - - - - - - - - -	- + - + - + - + - +	- 0 3 - - - 1 0 1 5	— 3
— 3	— 1 0 1 3	— 1 0 0 3	- + - + - + - + - +
- + - + - + - + - +	- 0 2 - - - 1 0 1 4	— 1 0 1 0	— 1 0 1 3
— 1 0 1 3	— 1 0 0 5	- 0 1 - - - 1 0 1 2	- 0 2 - - - 1 0 1 4
- 0 2 1 0 1 4	— 1 0 1 0	- - - - - - - - - -	- + - + - + - + - +
— 1 0 0 5	- 0 1 - - - 1 0 1 2	— 3	— 1 0 1 5
— 1 0 1 0	- - - - - - - - - -	+ + - + - + - + - +	- 0 3 - - - 1 0 1 5
- 0 1 - - - 1 0 1 2	— 3	— 1 0 1 3	- + - + - + - + - +
- - - - - - - - - -	- + - + - + - + - +	- 0 2 - - - 1 0 1 4	— 1 0 1 6
— 3	— 1 0 1 3	— 1 0 0 5	+ 1 0 1 6

Фиг. 6

Аналогични програми за съставяне на блок-схема и за проследяване са направени и за проверка на програми, написани на езика за символично програмиране на машината „Минск-2“ [2]. В тези програми вместо адреси от оперативната памет се отпечатват етикети. Подробни таблици за вида на индикациите се дава в [5]. Програмата „БС“ заема 603₍₈₎ клетки от оперативната памет, а програмата „Проследяване“ — 561₍₈₎.

ЛИТЕРАТУРА

1. Богданов, Д. И., П. Х. Бърнев, В. И. Василев, Бл. Х. Сендов. Електронни сметачни машини (с описание и инструкция за програмиране на „Минск-2“). С., 1966.
2. Инструкция за ползуване на системата за символично програмиране за машината „Минск-2“, О-ЧМ-20. С., 1967. Архив на Мат. инст. с Изчисл. център при БАН.
3. Документация на диспечерска система МИД-2. Ръкопис. С., 1968. Архив на Мат. инст. с Изчисл. център при БАН.
4. Компилираща система МИКС за използване на библиотеката стандартни програми на „Минск-2“, О-ЧМ-11. С., 1967. Архив на Мат. инст. с Изчисл. център при БАН.
5. Ръководство по програмиране в системата МИД-2 (част I) О-ЧМ-44. С., 1969, Архив на Мат. инст. с. Изчисл. център при БАН.

Постъпила на 13. III. 1970 г.

ПРОВЕРКА ЛОГИЧЕСКОЙ СТРУКТУРЫ ПРОГРАММ ДЛЯ ЭВМ МИНСК-2

Петр Бырнев и Александр Ахегукян

(Резюме)

Рассматривается вопрос о проверке логической структуры программ, написанных на машинном языке Минск-2 (Минск-22) или на языке символического программирования для этой машины. Описывается метод проверки и две программы: „Блок-схема“ и „Слежение“, которыми этот метод реализуется.

A CHECK OF THE LOGICAL STRUCTURE OF PROGRAMMES WRITTEN FOR MINSK-2

Petâr Bârnev, Aleksandâr Ahegukjan

(Summary)

The problem for a check of the logical structure of programmes written in the machine language of Minsk-2 (Minsk-22) or in the language for symbolic programming for this machine is considered. A method for such a check and two programmes "Flow Chart" and "Programme traces routine" for its realization are described.