# Министерство образования и науки Российской Федерации Севастопольский государственный университет

Кафедра ИС

#### Отчет

По дисциплине: "Теоретические основы построения компиляторов"
Лабораторная работа №2
"ИССЛЕДОВАНИЕ СКАНЕРА ПРИ АНАЛИЗЕ ПРОСТЫХ ЯЗЫКОВЫХ КОНСТРУКЦИЙ"

Выполнил:

ст.гр. ИС/б-20-1-о

Галенин А. К.

Проверил:

Карлусов В.Ю.

Севастополь

#### 1 ЦЕЛЬ РАБОТЫ

Изучить принципы построения и программирования лексического анализатора на языке С (С++) для простых языковых конструкций. Получить навыки практического построения лексического анализатора (сканера) на основе теории конечных автоматов. Освоить приёмы составления регулярных выражений для описания лексем. Закрепить навыки построения минимального КА, осуществляющего сканирование текста программ.

## 2 ПОСТАНОВКА ЗАДАЧИ

## Вариант 11 – 1.1.15

Таблица 1 – Типы идентификаторов

№ ва-	Описание типа
рианта	
1	Содержит чередующиеся пары букв и цифр, заканчивается последова-
	тельностью символов "1"

Таблица 2 – Типы констант

№ ва-	Тип	Пояснения	Формат
рианта			
1	F	С фиксированной точкой	±000.000

Таблица 3 – Фрагмент программы для анализа

No	Служебные слова	Раздел	пители	Фрагмент программы для анализа
ва		одноли	двули-	
p.		терные	терные	
15	WAIT	(	>=	WAIT(S)
	SIGNAL	)	<=	K=K+5.3 %
		>		C=K+C+1 %
		=		SIGNAL(S<=0xf)
		+		
		%		

## 3 ХОД РАБОТЫ

1. Были назначены коды лексемам:

WAIT - 100

SIGNAL - 200

Идентификатор, <iden>-300

Константа, <data> -400

(-501

) - 502

> -503

= -504

+ - 505

% - 506

>= - 507

<= - 508

2. Построение минимального конечного детерминированного автомата:

- 1) Служебные слова являются самоопределяющимися цепочками, составленными из литер {W, A, I, T S, G, N, L}
  - 2) Эскизно идентификатор (переменная) будет выглядеть так:

Где Б – любая латинская буква, Ц – любая десятичная цифра, 1 – последовательность 1, которой по условию задания оканчивается имя переменной.

3) Константа с фиксированной точкой описывается следующим образом:

$$(+v-)$$
ЦЦЦ  $\bullet$  ЦЦЦ

Где Ц – любая десятичная цифра

4) Множество букв =  $\{W, A, I, T, S, G, N, L, \delta\}$ 

где  $\delta$  – буква латинского алфавита, не совпадающая по начертанию с W, A, I, T, S, G, N, L.

- 5) Множество цифр  $\coprod = \{0, 2, 3, 4, 5, 6, 7, 8, 9\}$
- 6) Множество «не буква, не цифра»  $L_1 = \{(, ), >, =, +, \%, >=, <=, \bullet\} \cup L_3$  включает, помимо однолитерных разделителей и компонентов двулитерных разделителей, точку, отделяющую целую часть числа от дробной части, и любой символ, не принадлежащий алфавиту.
- 7) Множество  $L_2 = L_1$  текущая литера не принадлежит к образующим константу или литерал.
  - 8) L3 литера, не принадлежащая алфавиту конечного автомата.

Окончательно получаем дизъюнктивные члены выражения, описывающие конечный автомат:

- 1) WAITL $_1$  первое служебное слово
- 2) SIGNALL $_1$  второе служебное слово
- 3) (W V A V I V T V S V G V N V L V  $\delta$  V Ц) {W V A V I V T V S V G V N V L V  $\delta$  V Ц}  $1L_1$  переменная
  - 4) (+V-)ЦЦЦ ЦЦЦ $L_2$  константа
- 5) ( V ) V > V = V + V % V >= V <= V  $L_3$  однолитерные и двулитерные разделители

3. На основании полученного регулярного выражения была выполнена разметка.

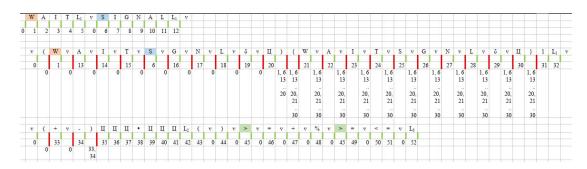


Рисунок 1 – Разметка регулярного выражения

4. На основе разметки была построена таблица ДКА и произведена её минимизация.

			Фу	нкция	пере	еход	OB,	ДКА	λ, п	остр	оен	ная	ю м	ини	ми	цир	ова	нно	му	pery	пярі	IOM	у ві	ыра	кен	ию					1	1							L	L			1											1			1
	0	1	2	3	4 :	5	6	7	8	9	10	11	12	13	3 :	4	15	16	17	18	15	2	0	21	22	23	24	25	26	5 2	7 :	28	29	30	31	32	33	34	35	30	5 3	7 3	88	39	40	41	42	43	44	1 4:	44	5 4	7 4	8	49	50	51
1	1	21				- 2	1	П						2		21	21	21	21	21	21	. 2	1	21	21	21	21	21	21	1 2	1 :	21	21	21					Т	Т	Т	$\top$	$\neg$							Т	Т		Т	T			
1	13	2				2	12	$\neg$		10	П	П	1	23	2 :	12	22	22	22	22	22	2	2	22	22	22	22	22	22	2 2	2 :	22	22	22					П	Т	Т	Т	Т												Г		
1	14	23	3		7	- 1	7						1	23	3 2	23	23	23	23	23	23	2	3 .	23	23	23	23	23	23	3 2	3 :	23	23	23								Т	$\neg$												Г		
1	15	24		4		2	.4	$\neg$					1	24		24	24	24	24	24	24	2	4	24	24	24	24	24	24	1 2	4 :	24	24	24					П	Т	Т	Т	Т	$\neg$											Г		
	6	25				2	25						1.	20	5 2		25	25	25	25	25		5	25	25	25	25	25	25		5 :	25	25	25	3							3	3												Г		
1	16	26		$\neg$		. 🗆	6	8					7	20	5 :	26	26	26	26	26	26	2	6	26	26	26	26	26	26	5 2	6 :	26	26	26	<u> </u>				П	Т			П.												П		
1	17	27			WAIT	2	.7		9				SIGN	2	7 2	27	27	27	27	27	27	2	7	27	27	27	27	27	27		7 :	27	27	27																					Г		. 1
1	18	28					18	П			11		18	21		28	28	28	28	28	28	2	8	28	28	28	28	28	28	3 2	8 :	28	28	28		атор			П	Т	Т		1												Г		
1	19	29			- 6	2	19						0	29	) (	9	29	29	29	29	29	2	9	29	29	29	29	29	25	2	9 .	29	29	29		E											E	L	_	١.	١.	١.	1	٠.	. [		
		31			- 610		1	$\neg$				П	02010	31		1	31	31	31	31	31	3	1	31	31	31	31	31	31	1 3	1 :	31	31	31		MX			П	Т	Т		$\top$				ž					+			ň. [		Ÿ.
2	20	30			-	3	0							3(	) :	10	30	30	30	30	30	3	0	30	30	30	30	30	30	0 3	0 :	30	30	30		ентиф	35	35	36	3	7	3	39 .	40	41		Константа	Знак	Знак	Знак	Знак	Знак	Знах		Знах		Знак
		32			5		Т	$\neg$				12	8	Г	Т		$\neg$			П	Т	Т	Т					П		т		Т		$\neg$	32	8			П	Т	Т		Т	$\neg$		42	ತಿ	1.0	1.,	m	m	m	m	, (	5		e l
4	44	32			5							12	6630													);				Т	$\top$				32	2				T		9				42									Г		
4	45	32		$\neg$	5 5 5 5 5 7	: F	Т	$\neg$				12		Г	Т		$\neg$			Т		Т	Т	$\neg$				П		т				$\neg$	32	_			П	Т			Т	П		42									Г		
4	46	32			5 0							12	S.									T								T					32											42										51	
4	47	32			5		Т	$\neg$				12	10	Г	Т		$\neg$			Т	Т	Т	Т	$\neg$				П		т	Т			$\neg$	32				П	Т			П			42									Г		
4	48	32			5			$\neg$				12	1		Т					Т		Т								T	Т				32											42									Г		
	T	$\neg$		$\neg$	5		T	$\neg$			П	12	1		Т	$\top$	$\neg$			Т	Т	Т	$\top$						1	T	T	$\neg$		$\neg$	32				Т	Т	3	8	$\neg$	$\neg$	$\neg$	42		1							- 1		
1 5	50	32			5		_					12	1			_						T	1					-	1	$\top$	Ť				32			l –			1	1	$\top$		$\neg$	42		1							- 1		
		32		-	5	$\perp$	$^{+}$	$\neg$			-	12	1		+	$^{+}$	$\neg$			-	+	+	+	$\rightarrow$		7	-	+	_	+	+	$^{+}$	$\neg$	$\neg$	32				_	-		+	$\pm$	$\rightarrow$		42		1		- 1		- 1			- 1		

Рисунок 2 – Функция переходов ДКА

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
W	1	11	802	802	805	11	805	805	805	805	805	11	802	803	803	803	803	803	803	803	803	804
A	11	2	802	802	805	11	805	805	9	805	805	11	802	803	803	803	803	803	803	803	803	804
I	11	11	3	802	805	6	805	805	805	805	805	11	802	803	803	803	803	803	803	803	803	804
T	11	11	802	4	805	11	805	805	805	805	805	11	802	803	803	803	803	803	803	803	803	804
S	5	11	802	802	805	11	805	805	805	805	805	11	802	803	803	803	803	803	803	803	803	804
G	11	11	802	802	805	11	7	805	805	805	805	11	802	803	803	803	803	803	803	803	803	804
N	11	11	802	802	805	11	805	8	805	805	805	11	802	803	803	803	803	803	803	803	803	804
L	11	11	802	802	805	11	805	805	805	10	805	11	802	803	803	803	803	803	803	803	803	804
õ	11	11	802	802	805	11	805	805	805	805	805	11	802	803	803	803	803	803	803	803	803	804
1	801	12	802	802	805	12	805	805	805	805	805	12	802	803	803	803	803	803	803	803	803	804
Ц	11	11	802	802	805	11	805	805	805	805	805	11	802	14	15	16	803	18	19	20	803	804
(	501	300	802	802	100	802	805	802	802	802	200	802	300	803	803	803	803	803	803	803	400	804
)	502	300	802	802	100	802	805	802	802	802	200	802	300	803	803	803	803	803	803	803	400	804
>	503	300	802	802	100	802	805	802	802	802	200	802	300	803	803	803	803	803	803	803	400	804
=	504	300	802	802	100	802	805	802	802	802	200	802	300	803	803	803	803	803	803	803	400	508
+	505	300	802	802	100	802	805	802	802	802	200	802	300	803	803	803	803	803	803	803	400	804
%	506	300	802	802	100	802	805	802	802	802	200	802	300	803	803	803	803	803	803	803	400	804
•	801	802	802	802	100	802	805	802	802	802	200	802	300	803	803	803	17	803	803	803	400	804
<	21	300	802	802	100	802	805	802	802	802	200	802	300	803	803	803	803	803	803	803	400	804
L <sub>3</sub>	600	300	802	802	100	802	805	802	802	802	200	802	300	803	803	803	803	803	803	803	400	804

Рисунок 3 – Минимизированная таблица переходов ДКА

#### 5. Код программы:

```
import java.io.FileReader;
import java.util.Scanner;
public class Main {
   private static int[][] matrix;
    private static Scanner scanner;
    private static boolean flag = false;
    private static int temp_ch;
    public static void main(String[] args) {
        try{
            int i, n;
            FileReader fr = new FileReader("data.txt");
            matrix = readMatrix(matrix); // управляющая таблица
            і=0; // номер состояния КА
            n=0; // число правильных слов
            String word = new String();
            int ch; // текущая литера
            ch = fr.read();
            while (ch != -1) {
                    if (Character.isLetter(ch)){
                        switch(ch) {
                             case 'W':
                                 i = matrix[0][i];
                                 word += (char) ch;
                                 break;
                             case 'A':
                                 i = matrix[1][i];
                                 word += (char) ch;
                                 break;
                             case 'I':
                                 i = matrix[2][i];
                                 word += (char) ch;
                                 break;
                             case 'T':
                                 i = matrix[3][i];
                                 word += (char) ch;
                                 break;
                             case 'S':
                                 i = matrix[4][i];
                                 word += (char) ch;
                                 break;
                             case 'G':
                                 i = matrix[5][i];
                                 word += (char) ch;
                                 break;
                             case 'N':
                                 i = matrix[6][i];
                                 word += (char) ch;
                                 break;
                             case 'L':
                                 i = matrix[7][i];
                                 word += (char) ch;
                                 break;
                             default:
                                 i = matrix[8][i];
                                 break;
                        }
                    else if (Character.isDigit(ch)) {
                                 i = matrix[9][i];
                    else {
```

```
switch (ch) {
                                   case '(':
                                       i = matrix[11][i];
                                       word += (char) ch;
                                       break;
                                    case ')':
                                        i = matrix[12][i];
                                       word += (char) ch;
                                       break;
                                    case '>':
                                        i = matrix[13][i];
                                       word += (char) ch;
                                       break;
                                    case '=':
                                       i = matrix[14][i];
                                       word += (char) ch;
                                       break;
                                    case '+':
                                       i = matrix[15][i];
                                       word += (char) ch;
                                       break;
                                    case '%':
                                       i = matrix[16][i];
                                       word += (char) ch;
                                       break;
                                    case '.':
                                       i = matrix[17][i];
                                       word += (char) ch;
                                       break;
                                    case '<':
                                       i = matrix[18][i];
                                       word += (char) ch;
                                       break;
                                    default:
                                        i = matrix[19][i];
                                       word += (char) ch;
                                       break;
                               }
                       if ((i > 99)&&(i < 500)) {
                           /* Анализ кода состояния */
                           switch(i){
                               case 100:
                                   System.out.println("Служебное слово 'WAIT'. Состояние =
"+i);
                                   break;
                               case 200:
                                   System.out.println("Служебное слово 'SIGNAL'. Состояние
= "+i);
                                   break;
                               case 300:
                                    System.out.println("Завершился идентификатор. Состояние
= "+i);
                                   break;
                               case 400:
                                    System.out.println("Завершилась константа. Состояние =
"+i);
                                   break;
                           }
                           i=0;
                           if (ch == '<' || ch == '=' || ch == '>') flag = true;
                       else if (i > 500 && i < 800) {
                           switch(i) {
```

```
System.out.println("Разделитель '('. Состояние = " + i);
                         break;
                    case 502:
                    System.out.println("Разделитель ')'. Состояние = " + i);
                         break:
                    case 503:
                    System.out.println("Разделитель '>'. Состояние = " + i);
                         break;
                    case 504:
                    System.out.println("Разделитель '='. Состояние = " + i);
                         break;
                    case 505:
                    System.out.println("Разделитель '+'. Состояние = " + i);
                    case 506:
                    System.out.println("Разделитель '%'. Состояние = " + i);
                         break;
                    case 507:
                    System.out.println("Разделитель '>='. Состояние = " + i);
                    case 508:
                    System.out.println("Разделитель '<='. Состояние = " + i);
                         break;
                    default:
                         System.out.println("Ошибка "+i);
                i=0;
            else if (i > 800) {
                switch(i) {
                    case 801:
                         System.out.println("Неправильное начало");
                         break;
                    case 802:
                         System.out.println("Ошибка в служебном слове");
                         break;
                    case 803:
                 System.out.println("Ошибка в написании имени переменной");
                    case 804:
                         System.out.println("Ошибочная константа");
                    case 805:
                         System.out.println("Ошибка");
                    default:
                         System.out.println("Неизвестная входная литера");
                i=0;
                ch = fr.read();
            if (!flag){
                ch = fr.read();
            }else{
                flag = false;
            }
        }
    }
        catch (Exception e){
        e.printStackTrace();
    }
}
public static int[][] readMatrix(int mas[][]) {
    try{
        scanner = new Scanner(new FileReader("matrix.txt"));
```

case 501:

```
int n = scanner.nextInt();
             int m = scanner.nextInt();
             if (scanner.hasNextInt()){
                  mas = new int [n][m];
                  for (int i=0; i<n; i++) {
                       for (int j = 0; j < m; j++) {
                           mas[i][j] = scanner.nextInt();
                  }
         } catch (Exception e){
             e.printStackTrace();
         return mas;
    public static void Print(int[][] matrix) {
         for (int i = 0; i < matrix.length; i++) {</pre>
             for (int j = 0; j < matrix[0].length; j++) {
    System.out.print(matrix[i][j] + " ");</pre>
             System.out.println();
        System.out.println();
    }
}
```

Рисунок 4 – Результат выполнения программы

#### ВЫВОДЫ

В ходе выполнения лабораторной работы были изучены принципы построения и программирования лексического анализатора на языке С (С++) для простых языковых конструкций. Получены навыки практического построения лексического анализатора (сканера) на основе теории конечных автоматов. Освоены приёмы составления регулярных выражений для описания лексем. Закреплены навыки построения минимального КА, осуществляющего сканирование текста программ.