Построение лексического анализатора на основании автоматной грамматики

Цели

- Научитсья строить лексические анализаторы на основе автоматной грамматики

Порядок выполнения

1. Построить автоматную грамматику
2. Построить автомат
3. Привести к детерминированному автомату
4. Реализовать автомат программно на языке программирования
5. Написать отчет

Рекомендации по выполнению

- Массивы фиксированной длины
- Данные задаются внутри исходного кода

Состав отчета

- Титульный лист (фамилия, группа, номер варианта, наименование работы, задание)
- Текст задания
- Грамматика
- Диаграмма переходов
- Автоматы
- Текст программы

Варианты заданий В задании указано содержательное описание грамматики и простейши пример для облегчения понимания.

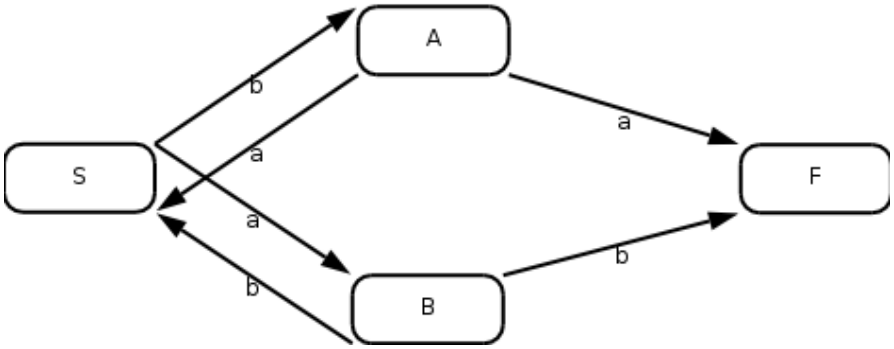
1. Одинаковые символы стоят парами: *aabbaabbbbaa*
2. В начале строки a: *ababbbabb*
3. Первый символ — не важен, далее одни a: *baaaaaaaaa*, *aaaaaaaaaa*
4. Либо одни a, либо одни b: *aaaaaaaaaa*, *bbbbbbbbb*
5. В конце строки b: *ababbbabb*
6. Одинаковые символы не должны стоять рядом: *abababababab*, *babababababa*
7. В строке должна встретиться хотя бы одна буква a: *bbbbbabbb*, *aaaaaaaaa*
8. Предпоследним символом строки должна быть b, *abbabaabb*
9. Вторым символом строки должна быть a: *baaaaabbb*
10. Два последних символа должны быть b: *abababb*, *bbbbbb*
11. Первый и третий символы должны быть разными: *aabbbabab*, *baaabbbab*
12. Первый и последний символы должны быть одинковыми: *ababababa*, *babbbabab*

Пример

Задание Символы a и b стоят парами: abbaabab, baabbaba.

Грамматика Построим грамматику по заданию:
$S \to aB$
$S \to bA$
$B \to bF$
$A \to aF$
$B \to bS$
$A \to aS$
$F \to \dashv$

Автомат Построим диаграмму переходов



	$-$	$-$	$-$	$+$
	S	A	B	F
a	B	S, F		
b	A		S, F	

Детерминированный автомат Перейдём к детерменированному автомату.

	$-$	$-$	$-$	$+$	$-$	$+$
	S	A	B	F	$\{\}$	$\{S, F\}$
a	B	$\{S, F\}$	$\{\}$	$\{\}$	$\{\}$	B
b	A	$\{\}$	$\{S, F\}$	$\{\}$	$\{\}$	A

Программа

```
#include <stdio.h>

#define S  0 /* S */
#define A  1 /* A */
#define B  2 /* B */
#define F  3 /* F */
#define H  4 /* {S,F} */
#define W  5 /* {} */

#define P  1 /* + */
#define M  0 /*   */

#define a 0 /* a */
#define b 1 /* b */

struct action
{
    int sos; /* состояние */
    int out; /* вывод */
};
```

```
int main()
{
    int table[2][6];
    table[a][S] = B;
    table[b][S] = A;
    table[a][A] = H;
    table[b][A] = W;
    table[a][B] = W;
    table[b][B] = H;
    table[a][F] = W;
    table[b][F] = W;
    table[a][W] = W;
    table[b][W] = W;
    table[a][H] = B;
    table[b][H] = A;

    int output[6];
    output[S] = M;
    output[A] = M;
    output[B] = M;
    output[F] = P;
    output[W] = M;
    output[H] = P;


    int input[]={b,a,a,b};
    int n = 4, i;
    int isos = S;

    for( i=0; i<n; ++i)
    {
        isos=table[input[i]][isos];
    }

    printf("%d\n", output[isos]);
    return 0;
}
```

Listing 1: анализатор