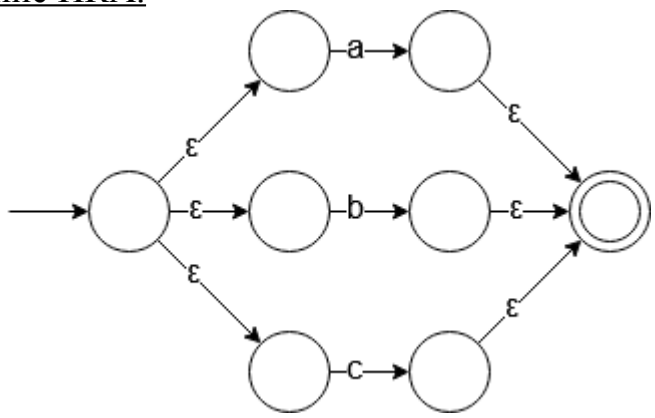
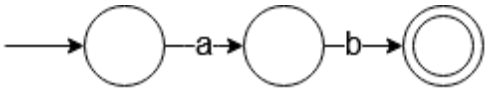
Исходное регулярное выражение: **ab(a|b|c)*ba**

Этап 1. Построение НКА.

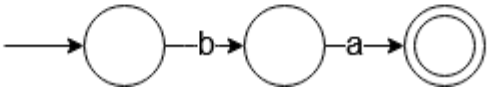
1 a|b|c



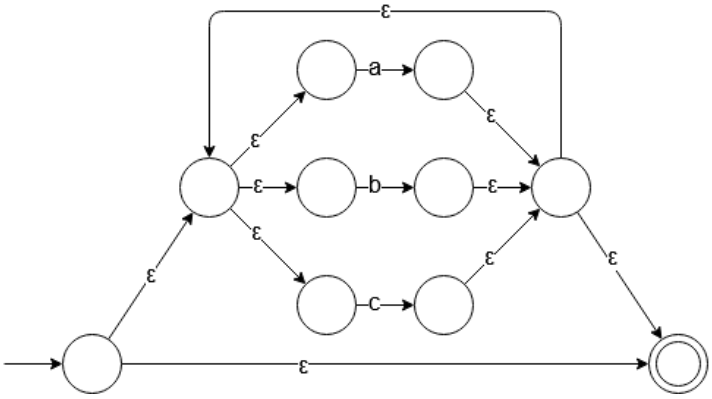
2 ab



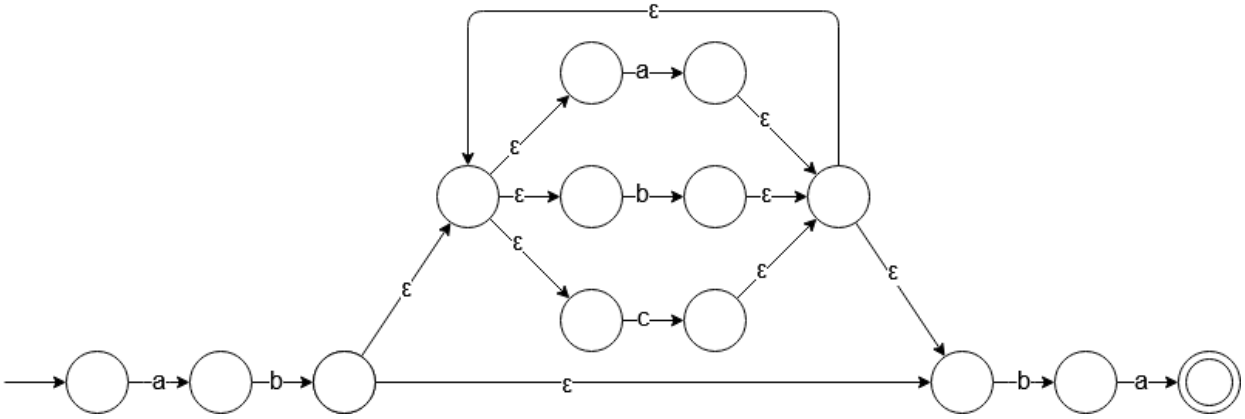
3 ba



4 (a|b|c)*

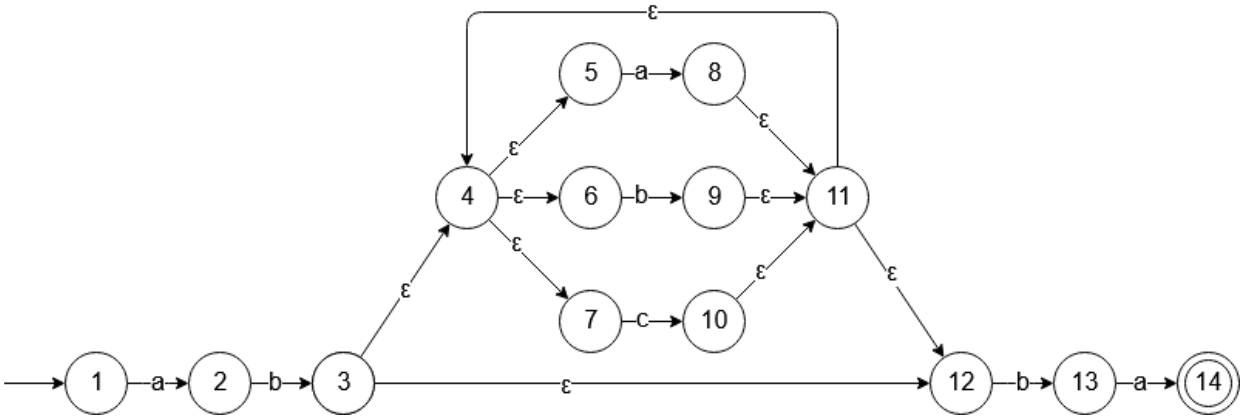


ab(a|b|c)*ba:



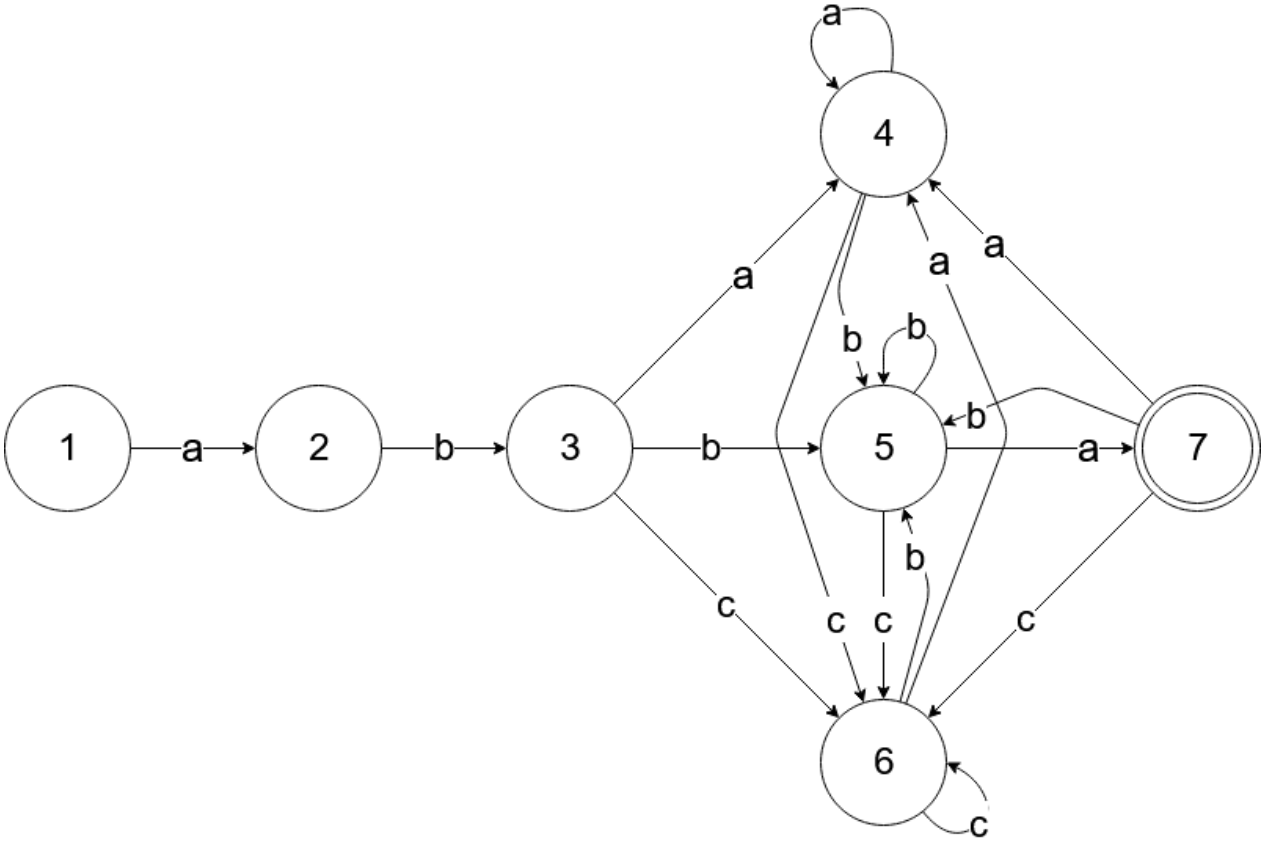
Этап 2. Построение ДКА по НКА.

Исходный НКА:



№	Состояние	a	b	c
1	1	2	-	-
2	2	-	3	-
3	3	4,5,6,7,8,11,12	4,5,6,7,9,11,12,13	4,5,6,7,10,11,12
4	4,5,6,7,8,11,12	4,5,6,7,8,11,12	4,5,6,7,9,11,12,13	4,5,6,7,10,11,12
5	4,5,6,7,9,11,12,13	4,5,6,7,8,11,14	4,5,6,7,9,11,12,13	4,5,6,7,10,11,12
6	4,5,6,7,10,11,12	4,5,6,7,8,11,12	4,5,6,7,9,11,12,13	4,5,6,7,10,11,12
7	4,5,6,7,8,11,14	4,5,6,7,8,11,12	4,5,6,7,9,11,12,13	4,5,6,7,10,11,12

Полученный ДКА:



Этап 3. Минимизация полученного ДКА.

Разделим состояния автомата:

P_0={ A_0={1,2,3,4,5,6}, B_0={7}}

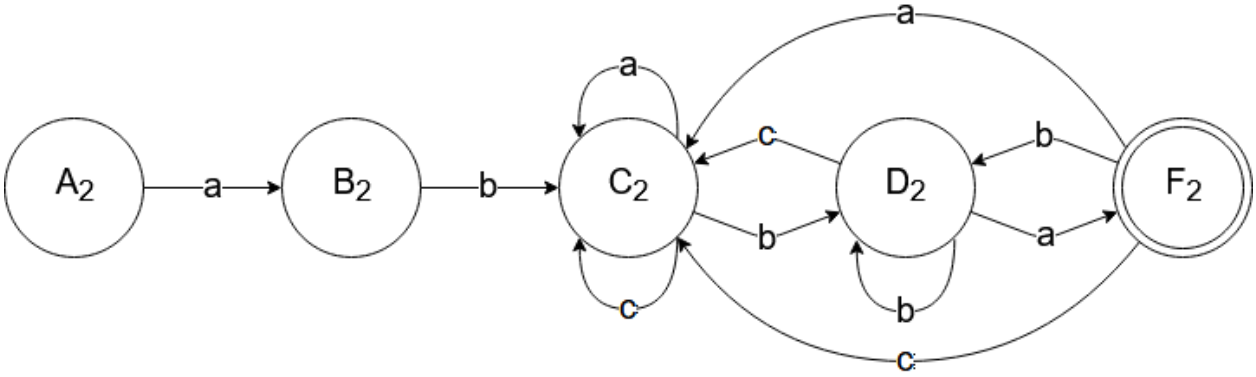
P_1={ A_1={1}, B_1={2}, C_1={3,4,6}, D_1={5}, F_1={7} }

P_2={ A_2={1}, B_2={2}, C_2={3,4,6}, D_2={5}, F_2={7} }

Построим таблицу переходов:

	a	b	c	a	b	c	a	b	c	a	b	c	
1	2	-	-	A_0	-	-	B_1	-	-	B_2	-	-	A
2	-	3	-	-	A_0	-	-	C_1	-	-	C2	-	B
3	4	5	6	A_0	A_0	A_0	C_1	D_1	C_1	C_2	D_2	C_2	C
4	4	5	6	A_0	A_0	A_0	C_1	D_1	C_1	C_2	D_2	C_2	C
5	7	5	6	B_0	A_0	A_0	F_1	D_1	C_1	F_2	D_2	C_2	D
6	4	5	6	A_0	A_0	A_0	C_1	D_1	C_1	C_2	D_2	C_2	C
7	4	5	6	A_0	A_0	A_0	C_1	D_1	C_1	C_2	D_2	C_2	F

Минимизированный ДКА:



Исходный код распознавателя:

```
#include <stdio.h>
#include <stdlib.h>

const int MAX_SIZE = 1024;

int check(int start_state, int start_sym_num, char * string)
{
	char c = string[start_sym_num];
	int state = start_state, i = start_sym_num;
	do
	{
		switch (state)
		{
			case 3:
				switch (c)
				{
					case 'a':
						break;

					case 'b':
						state = 4;
						break;

					case 'c':
```

```
                                        break;

                                default:
                                        return 0;
                        }
                        break;

                case 4:
                        switch (c)
                        {
                                case 'a':
                                        state = 5;
                                        break;

                                case 'b':
                                        break;

                                case 'c':
                                        state = 3;
                                        break;

                                default:
                                        return 0;
                        }
                        break;

                case 5:
                        switch (c)
                        {
                                case 'a':
                                        state = 3;
                                        break;

                                case 'b':
                                        state = 4;
                                        break;

                                case 'c':
                                        state = 3;
                                        break;

                                default:
                                        return 0;
                        }
                        break;
                }
                c = string[++i];
        }
        while(c != 0);

        if (state == 5)
                return 1;
        else
                return 0;
}

int main()
{
        char * string;
        string = (char*) malloc (MAX_SIZE);
        scanf("%s", string);

        if (string[0] == 'a' && string [1] == 'b' && check(3, 2, string))
                puts("Yes");
        else
                puts("No");
        free(string);
        return 0;
}
```