

## Exp 3 - Regular Expression to NFA

**Aim:** To convert a given regular expression (RE) into NFA.

### **Program:**

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <ctype.h>
int main()
{
    char m[20], t[10][10];
    int n, i, j, r = 0, c = 0;
    printf("REGULAR EXPRESSION TO NFA");
    for (i = 0; i < 10; i++)
    {
        for (j = 0; j < 10; j++)
        {
            t[i][j] = ' ';
        }
    }
    printf("\nEnter a regular expression:");
    scanf("%s", m);
    n = strlen(m);
    for (i = 0; i < n; i++)
    {
        switch (m[i])
        {
            case '|':
            {
                t[r][r + 1] = 'E';
                t[r + 1][r + 2] = m[i - 1];
                t[r + 2][r + 5] = 'E';
                t[r][r + 3] = 'E';
                t[r + 4][r + 5] = 'E';
                t[r + 3][r + 4] = m[i + 1];
                r = r + 5;
                break;
            }
            case '*':
            {
                t[r - 1][r] = 'E';

                t[r][r + 1] = 'E';
                t[r][r + 3] = 'E';
```

### Exp 3 - Regular Expression to NFA

RA1911003010143

Abhishek Kumar

```
t[r + 1][r + 2] = m[i - 1];
t[r + 2][r + 1] = 'E';
t[r + 2][r + 3] = 'E';
r = r + 3;
break;
}
case '+':
{
    t[r][r + 1] = m[i - 1];
    t[r + 1][r] = 'E';
    r = r + 1;
    break;
}
default:
{
    if (c == 0)
    {
        if ((isalpha(m[i])) && (isalpha(m[i + 1])))
        {
            t[r][r + 1] = m[i];
            t[r + 1][r + 2] = m[i + 1];
            r = r + 2;
            c = 1;
        }
        c = 1;
    }
    else if (c == 1)
    {
        if (isalpha(m[i + 1]))
        {
            t[r][r + 1] = m[i + 1];
            r = r + 1;
            c = 2;
        }
    }
    else
    {
        if (isalpha(m[i + 1]))
        {
            t[r][r + 1] = m[i + 1];
            r = r + 1;
            c = 3;
        }
    }
}
break;
}

}
printf("\n");
for (j = 0; j <= r; j++)
    printf(" %d", j);
printf("\n \n");
printf("\n");
for (i = 0; i <= r; i++)
{
    for (j = 0; j <= r; j++)
```

## Exp 3 - Regular Expression to NFA

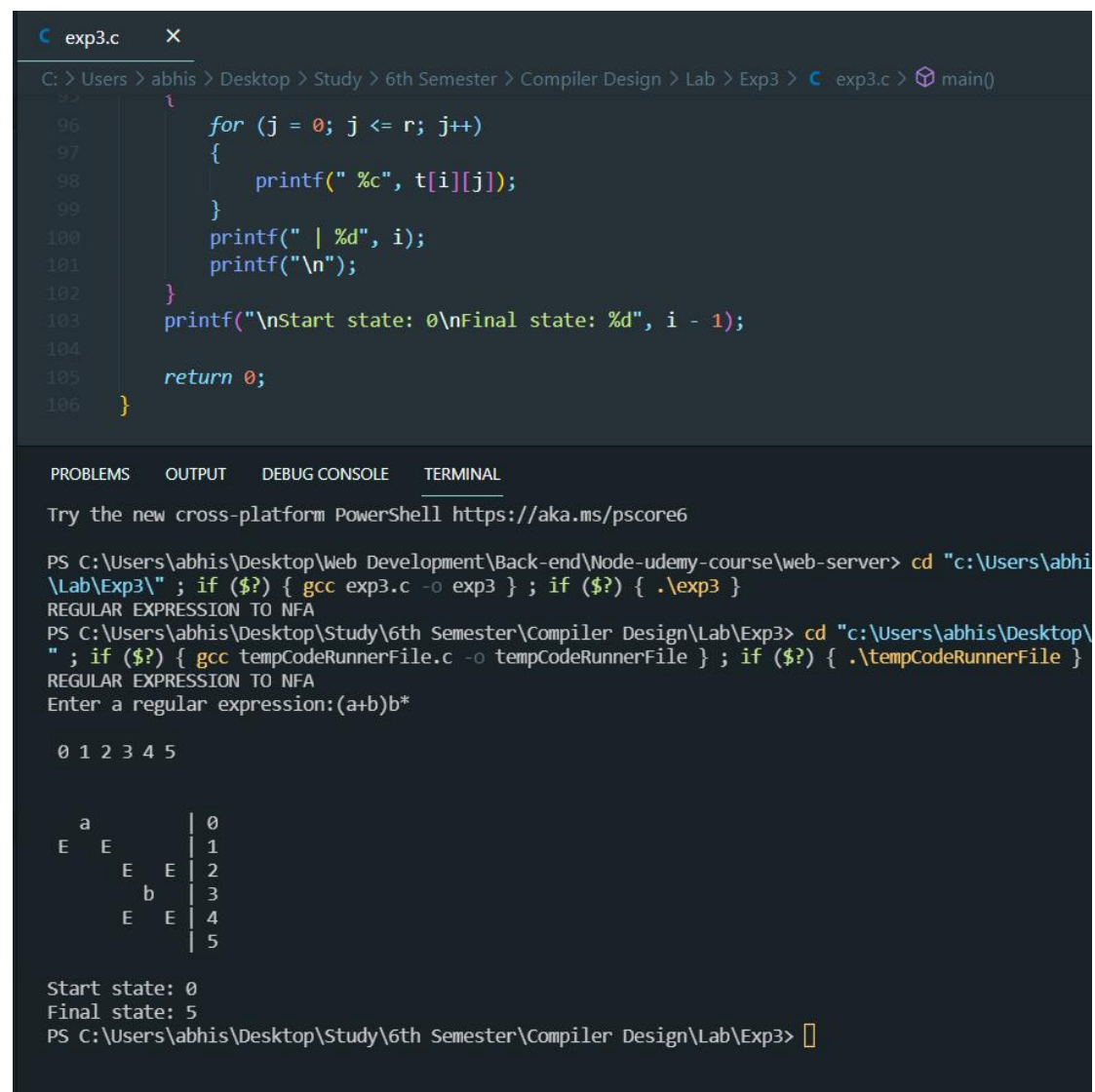
RA1911003010143

Abhishek Kumar

```
    {
        printf(" %c", t[i][j]);
    }
    printf(" | %d", i);
    printf("\n");
}
printf("\nStart state: 0\nFinal state: %d", i - 1);

return 0;
}
```

## Output:



The screenshot shows a code editor with a C program and a terminal window. The C program defines a function `main()` that takes a regular expression as input and prints the corresponding NFA states and transitions. The terminal output shows the execution of the program, including the input regular expression `(a+b)b*` and the resulting NFA diagram.

```
exp3.c X
C: > Users > abhis > Desktop > Study > 6th Semester > Compiler Design > Lab > Exp3 > exp3.c > main()
96     for (j = 0; j <= r; j++)
97     {
98         printf(" %c", t[i][j]);
99     }
100    printf(" | %d", i);
101    printf("\n");
102    }
103    printf("\nStart state: 0\nFinal state: %d", i - 1);
104
105    return 0;
106 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\Users\abhis\Desktop\Web Development\Back-end\Node-udemy-course\web-server> cd "c:\Users\abhis\Desktop\Study\6th Semester\Compiler Design\Lab\Exp3" ; if (\$?) { gcc exp3.c -o exp3 } ; if (\$?) { .\exp3 }

REGULAR EXPRESSION TO NFA

PS C:\Users\abhis\Desktop\Study\6th Semester\Compiler Design\Lab\Exp3> cd "c:\Users\abhis\Desktop\Study\6th Semester\Compiler Design\Lab\Exp3" ; if (\$?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if (\$?) { .\tempCodeRunnerFile }

REGULAR EXPRESSION TO NFA

Enter a regular expression:(a+b)b\*

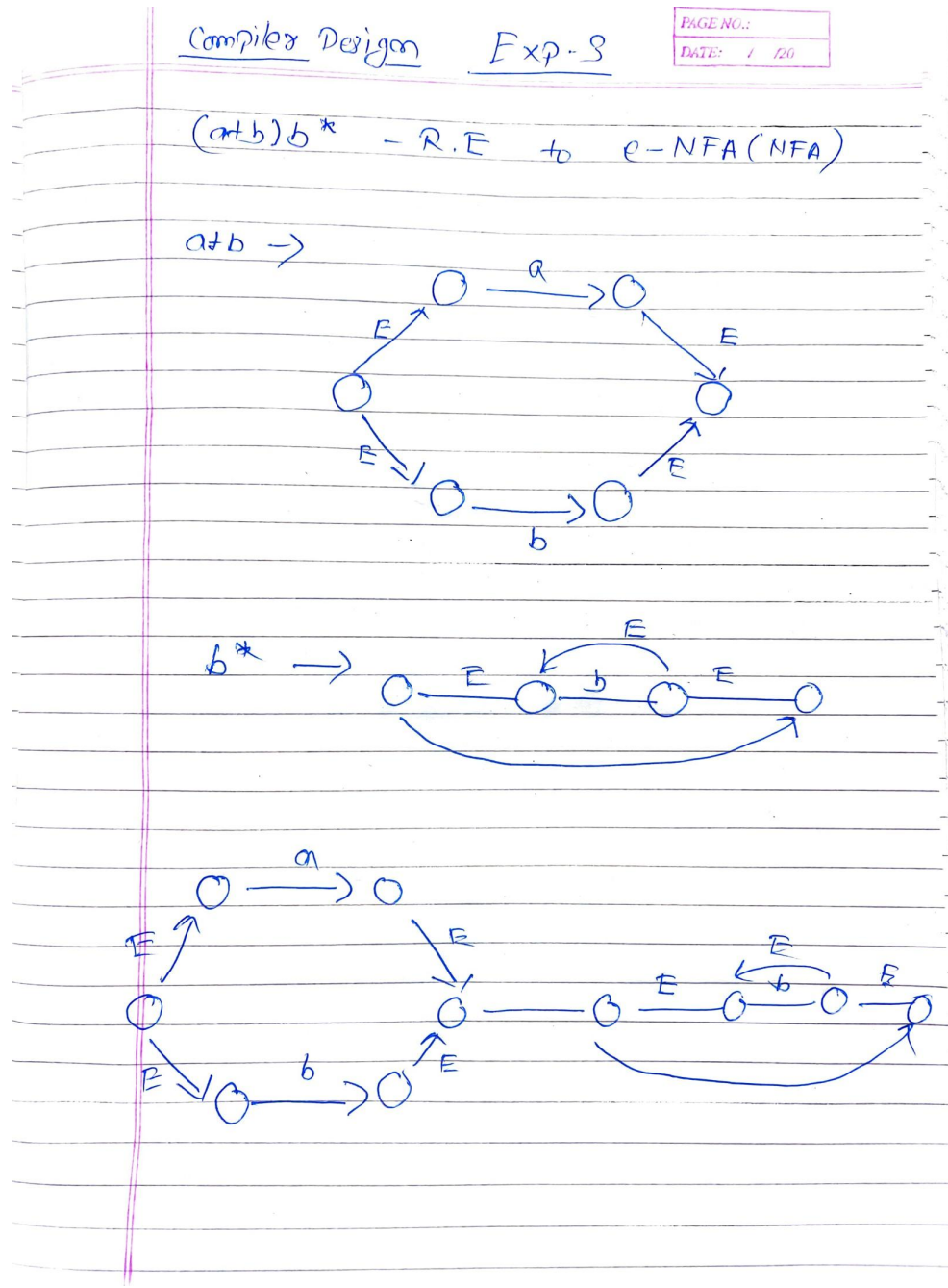
0 1 2 3 4 5

a					0
E	E				1
	E	E			2
		b			3
		E	E		4
					5

Start state: 0  
Final state: 5

PS C:\Users\abhis\Desktop\Study\6th Semester\Compiler Design\Lab\Exp3>

## Manual Calculation:



**Result:** Program to convert RE to NFA was written and executed successfully.