Compiler Design(18CSC304J)

Experiment 4

NFA to DFA

Harsh Goel
RA1811003010185

Aim: To study and perform NFA to DFA conversion

Language: C

Procedure:

1. Create a file or select the file for performing the operations on.
2. For this, created a c file called with extension as .c
3. Write the code in the c file
4. Run the code and perform the operations required.
5. Note the output and document it.
6. The NFA to DFA conversion will be verified

Process:

- Open the c file using open command in execute mode
- Declare the NFA expression using 2D array of int type
- Initialize the DFA expression with 1st variables same as NFA (initial value)
- Loop on the NFA till its length and perform conversion
- Print the DFA expression.

Code Snippet:

```c
#include <stdio.h>
int main()
{
   int nfa[5][2];
    nfa[1][1]=12;
    nfa[1][2]=1;
    nfa[2][1]=0;
    nfa[2][2]=3;
    nfa[3][1]=0;
    nfa[3][2]=4;
    nfa[4][1]=0;
    nfa[4][2]=0;
    int dfa[10][2];
    int dstate[10];
    int i=1,n,j,k,flag=0,m,q,r;
```

```
    dstate[i++]=1;
    n=i;
    // Init DFA initial Stages
    dfa[1][1]=nfa[1][1];
    dfa[1][2]=nfa[1][2];
    // Print dfa initial
    printf("\nf(%d,a)=%d",dstate[1],dfa[1][1]);
    printf("\nf(%d,b)=%d",dstate[1],dfa[1][2]);

// Loop over the dfa
 for(j=1;j<n;j++)
    {
        if(dfa[1][1]!=dstate[j])
          flag++;
    }
    if(flag==n-1)
    {
        dstate[i++]=dfa[1][1];
        n++;
    }
    flag=0;
    for(j=1;j<n;j++)
    {
         if(dfa[1][2]!=dstate[j])
            flag++;
    }
    if(flag==n-1)
    {
         dstate[i++]=dfa[1][2];
         n++;
    }
    k=2;
    while(dstate[k]!=0)
    {
        m=dstate[k];
        if(m>10)
        {
            q=m/10;
            r=m%10;
        }
        if(nfa[r][1]!=0)
             dfa[k][1]=nfa[q][1]*10+nfa[r][1];
        else
            dfa[k][1]=nfa[q][1];
        if(nfa[r][2]!=0)
            dfa[k][2]=nfa[q][2]*10+nfa[r][2];
        else
            dfa[k][2]=nfa[q][2];
```

```
        printf("\nf(%d,a)=%d",dstate[k],dfa[k][1]);
        printf("\nf(%d,b)=%d",dstate[k],dfa[k][2]);

       flag=0;
        for(j=1;j<n;j++)
        {
         if(dfa[k][1]!=dstate[j])
          flag++;
        }
      if(flag==n-1)
     {
        dstate[i++]=dfa[k][1];
        n++;
     }
    flag=0;
    for(j=1;j<n;j++)
    {
         if(dfa[k][2]!=dstate[j])
            flag++;
    }
    if(flag==n-1)
    {
         dstate[i++]=dfa[k][2];
         n++;
    }
    k++;
    }
    return 0;
}
```

Output Screenshots:

```
f(14,b)=1
PS C:\Users\HARSH-PC\Desktop\college\COMPILER_DESIGN\exp_4> ./master

f(1,a)=12
f(1,b)=1
f(12,a)=12
f(12,b)=13
f(13,a)=12
f(13,b)=14
f(14,a)=12
f(14,b)=1
PS C:\Users\HARSH-PC\Desktop\college\COMPILER_DESIGN\exp_4>
```

Result:

The code was successfully implemented in C and output was recorded. Hence, NFA to DFA was successfully executed.