Regular Expression to E-NFA (Thomson's Construction Method)

#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;

dfa[1][1]=nfa[1][1];

dfa[1][2]=nfa[1][2];

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

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

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:

