EXPERIMENT 7

AIM

DFA minimization using C program.

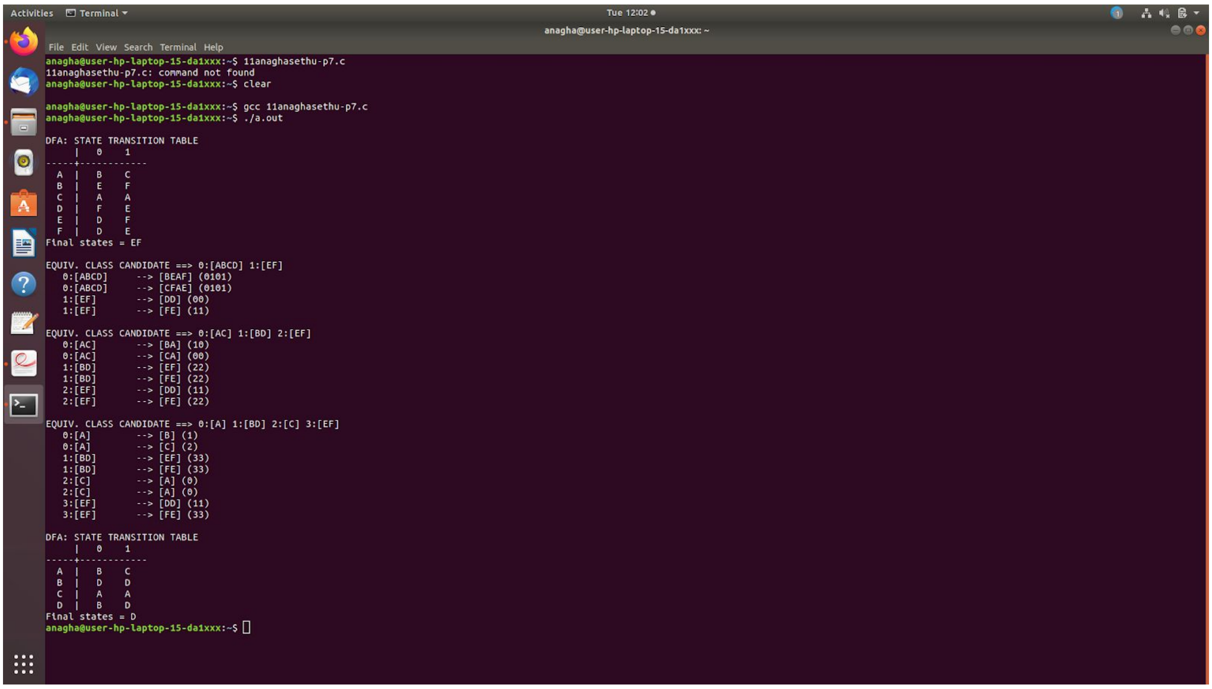
ALGORITHM

1. Start
2. Divide Q (set of states) into two sets. One set will contain all final states and the other set will contain non-final states. This partition is called P_0.
3. Initialize k = 1
4. Find P_k by partitioning the different sets of P_{k-1}. In each set of P_{k-1}, we will take all possible pair of states. If two states of a set are distinguishable, we will split the sets into different sets in P_k.
5. Stop when $P_k = P_{k-1}$ (No change in partition)
6. All states of one set are merged into one. No. of states in minimized DFA will be equal to no. of sets in P_k.
7. Stop

OUTPUT

gcc 11anaghasethu-p7.c

./a.out



```
anagha@user-hp-laptop-15-da1xxx:~$ 11anaghasethu-p7.c
11anaghasethu-p7.c: command not found
anagha@user-hp-laptop-15-da1xxx:~$ clear

anagha@user-hp-laptop-15-da1xxx:~$ gcc 11anaghasethu-p7.c
anagha@user-hp-laptop-15-da1xxx:~$ ./a.out

DFA: STATE TRANSITION TABLE
     |   0    1
-----+------------
  A  |   B    C
  B  |   E    F
  C  |   A    A
  D  |   F    E
  E  |   D    F
  F  |   D    E
Final states = EF

EQUIV. CLASS CANDIDATE ==> 0:[ABCD] 1:[EF]
   0:[ABCD]     --> [BEAF] (0101)
   0:[ABCD]     --> [CFAE] (0101)
   1:[EF]       --> [DD] (00)
   1:[EF]       --> [FE] (11)

EQUIV. CLASS CANDIDATE ==> 0:[AC] 1:[BD] 2:[EF]
   0:[AC]       --> [BA] (10)
   0:[AC]       --> [CA] (00)
   1:[BD]       --> [EF] (22)
   1:[BD]       --> [FE] (22)
   2:[EF]       --> [DD] (11)
   2:[EF]       --> [FE] (22)

EQUIV. CLASS CANDIDATE ==> 0:[A] 1:[BD] 2:[C] 3:[EF]
   0:[A]        --> [B] (1)
   0:[A]        --> [C] (2)
   1:[BD]       --> [EF] (33)
   1:[BD]       --> [FE] (33)
   2:[C]        --> [A] (0)
   2:[C]        --> [A] (0)
   3:[EF]       --> [DD] (11)
   3:[EF]       --> [FE] (33)

DFA: STATE TRANSITION TABLE
     |   0    1
-----+------------
  A  |   B    C
  B  |   D    D
  C  |   A    A
  D  |   B    D
Final states = D
anagha@user-hp-laptop-15-da1xxx:~$ 
```