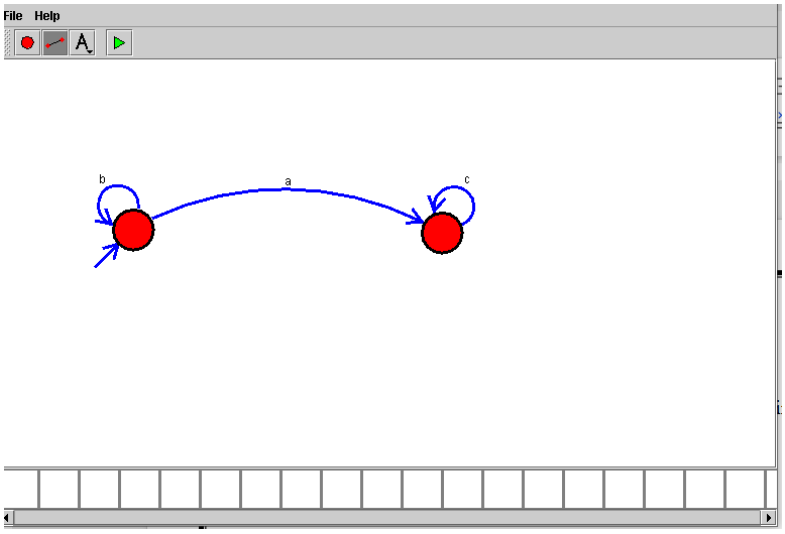
EXP :9

AIM: Design DFA using simulator to accept the input string “a” ,”ac”,and ”bac”

Simulaton



EXP :10

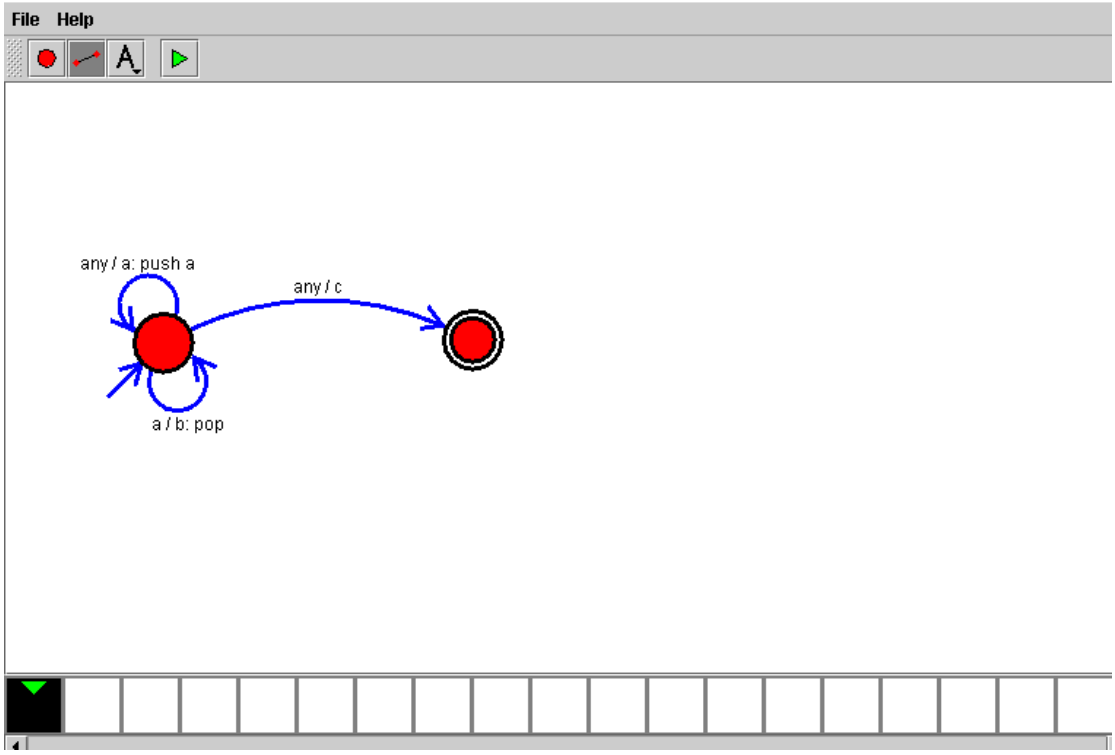
AIM: Design a Push Down Automata that accepts the language

$$L = \{w \mid w \in (a+b)* \; and \; n_a(w) = n_b(w)\}$$

$n_a(w)$ is the number of a's in w

$n_b(w)$ is the number of b's in w

Simulation



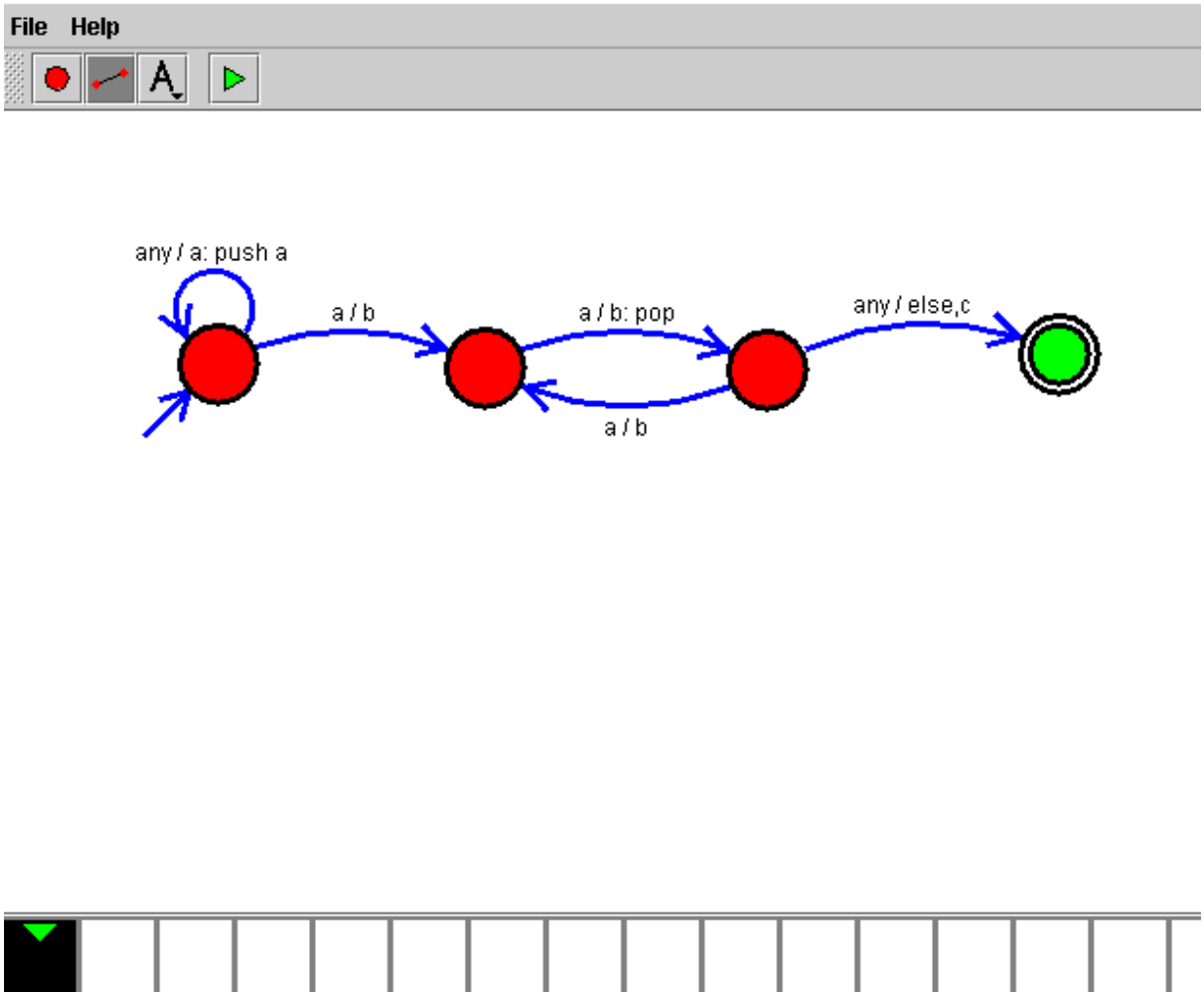
1.

EXP :11

AIM:Design PDA using simulator to accept the input string a^nb^{2n}

$$L = \{ a^n b^{2n} \mid w \in (a + b)$$

Simulation:

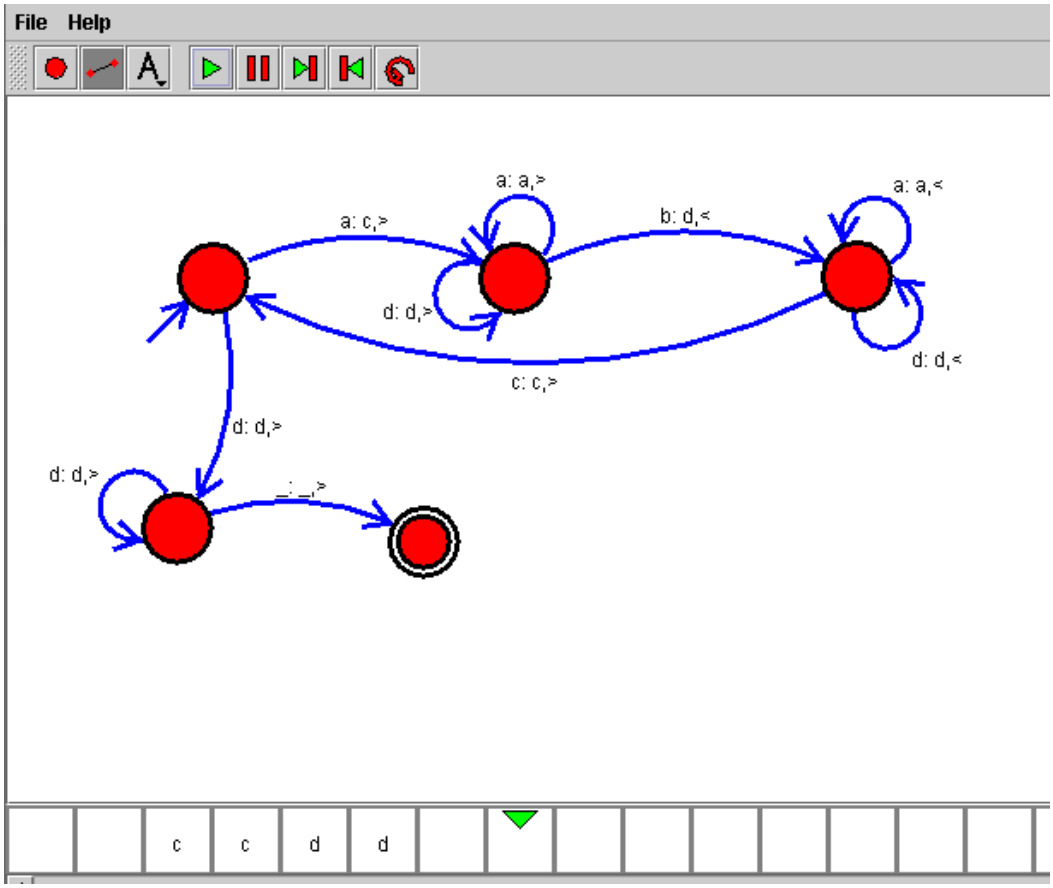


EXP :12

AIM:Design TM using simulator to accept the input string a^nb^n

$$L = \{ a^n b^n \mid w \in (a + b)$$

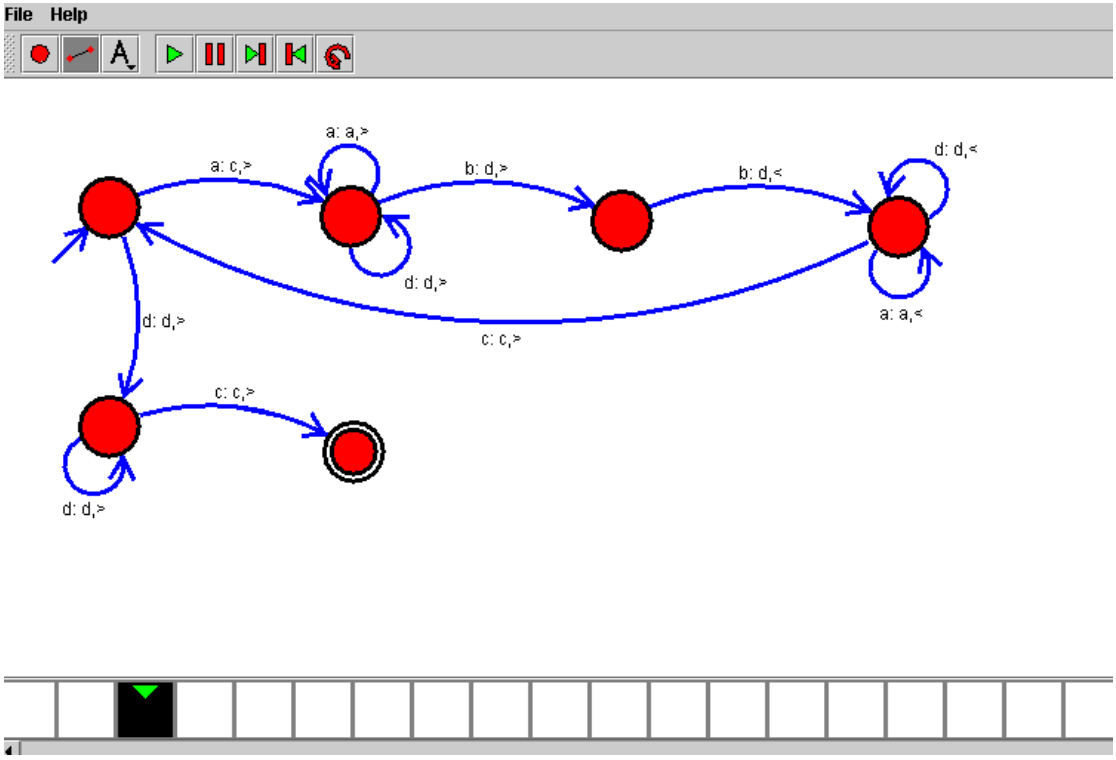
Simulation



EXP :13

AIM:Design TM using simulator to accept the input string $a^n b^{2n}$

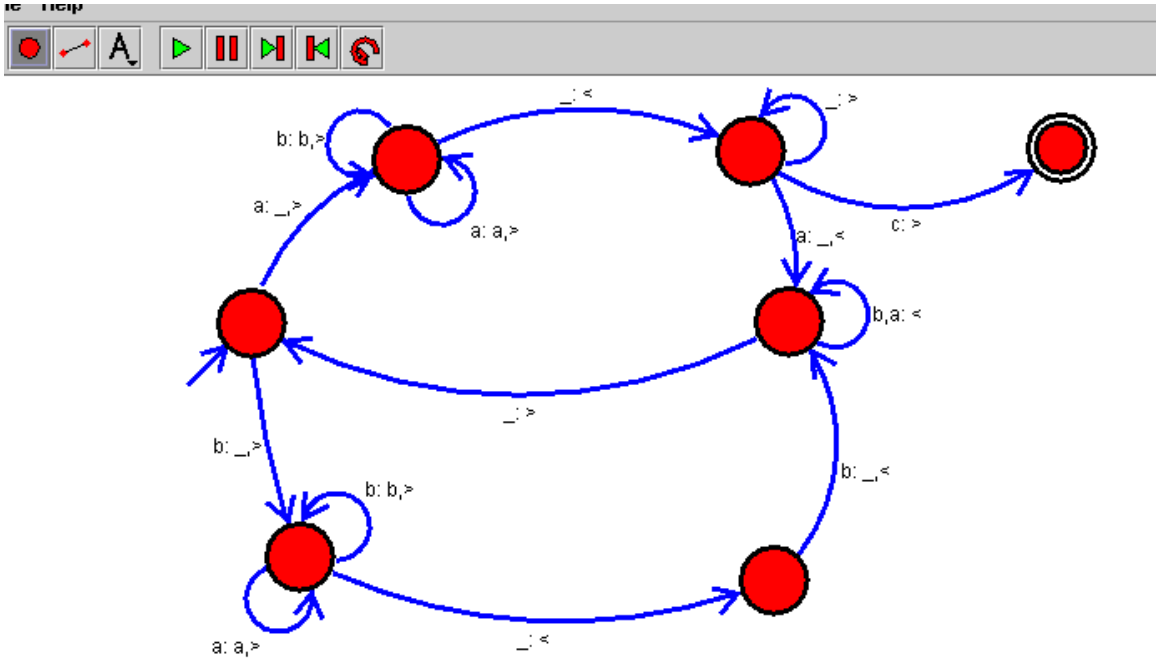
Simulation:



EXP :14

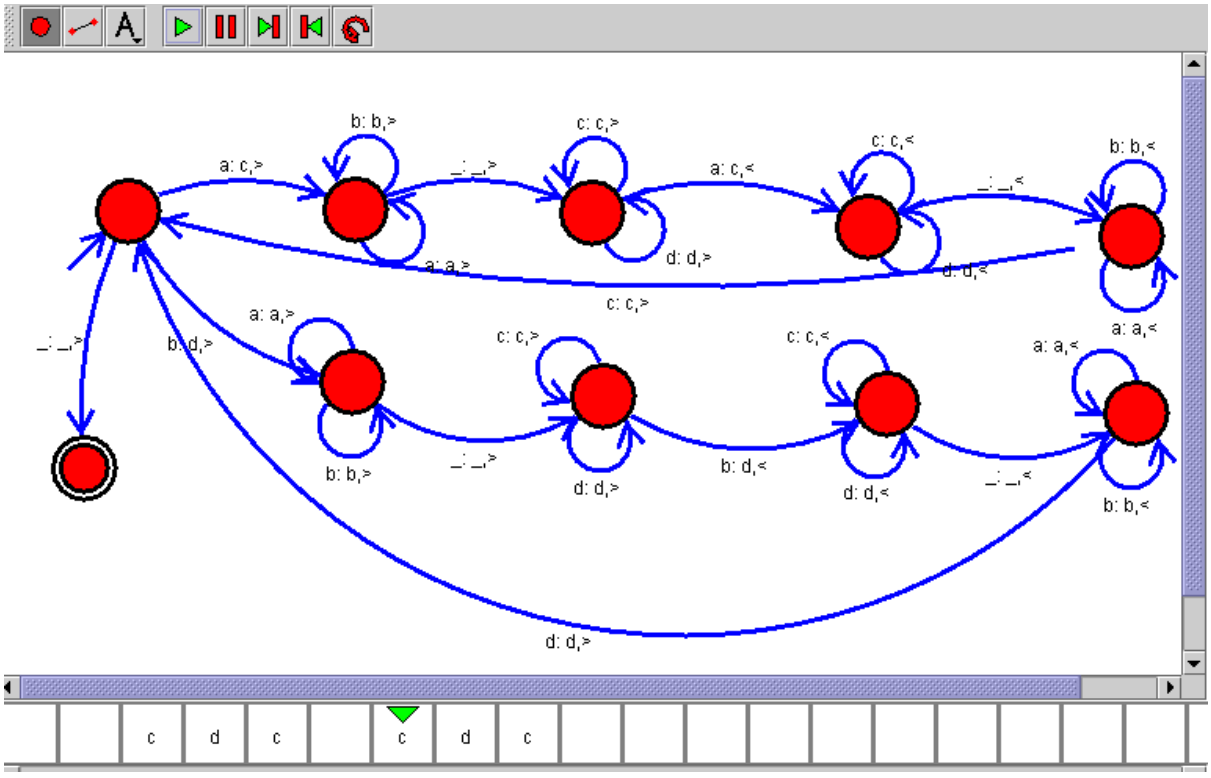
AIM:Design TM using simulator to accept the input string Palindrome ababa

Simulation:



EXP :15

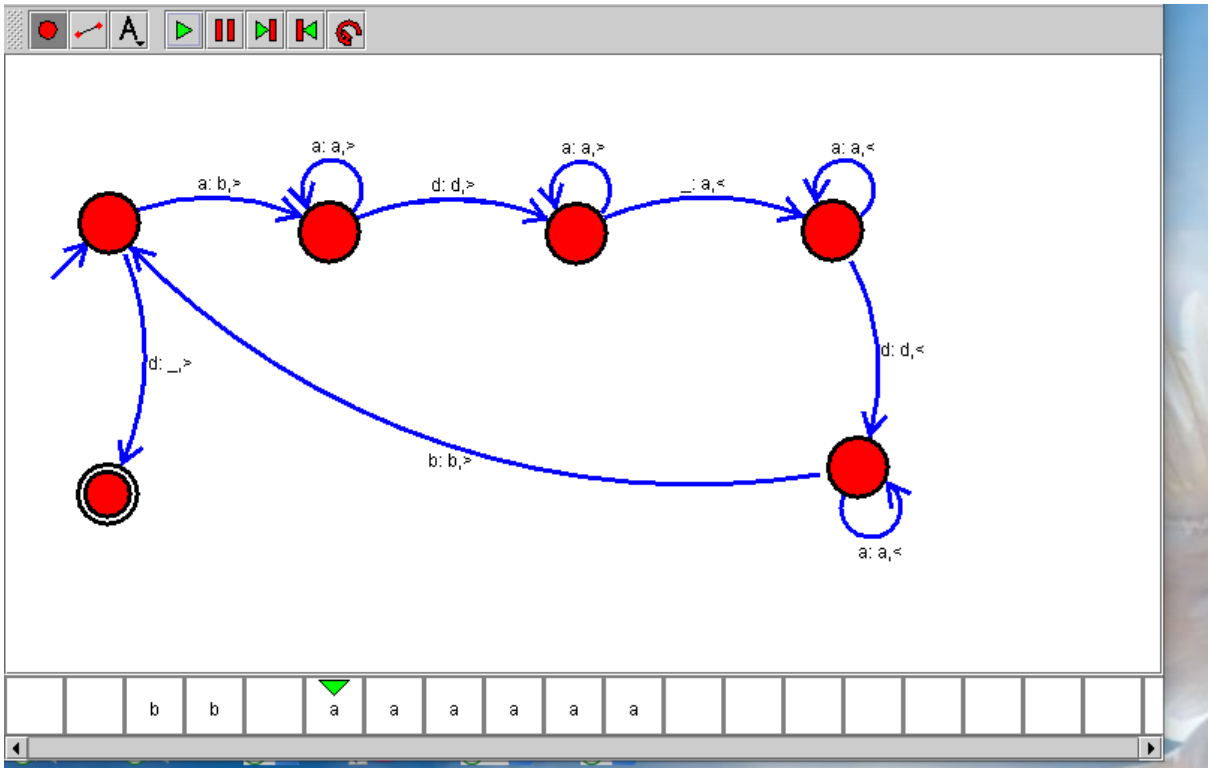
AIM: Design TM using simulator to accept the input string ww



EXP :16

AIM: Design TM using simulator to perform addition of ‘aa’ and ‘aaa’

W= aa+ aaaa
After Addition of a’s = aaaaaa
Simulation:



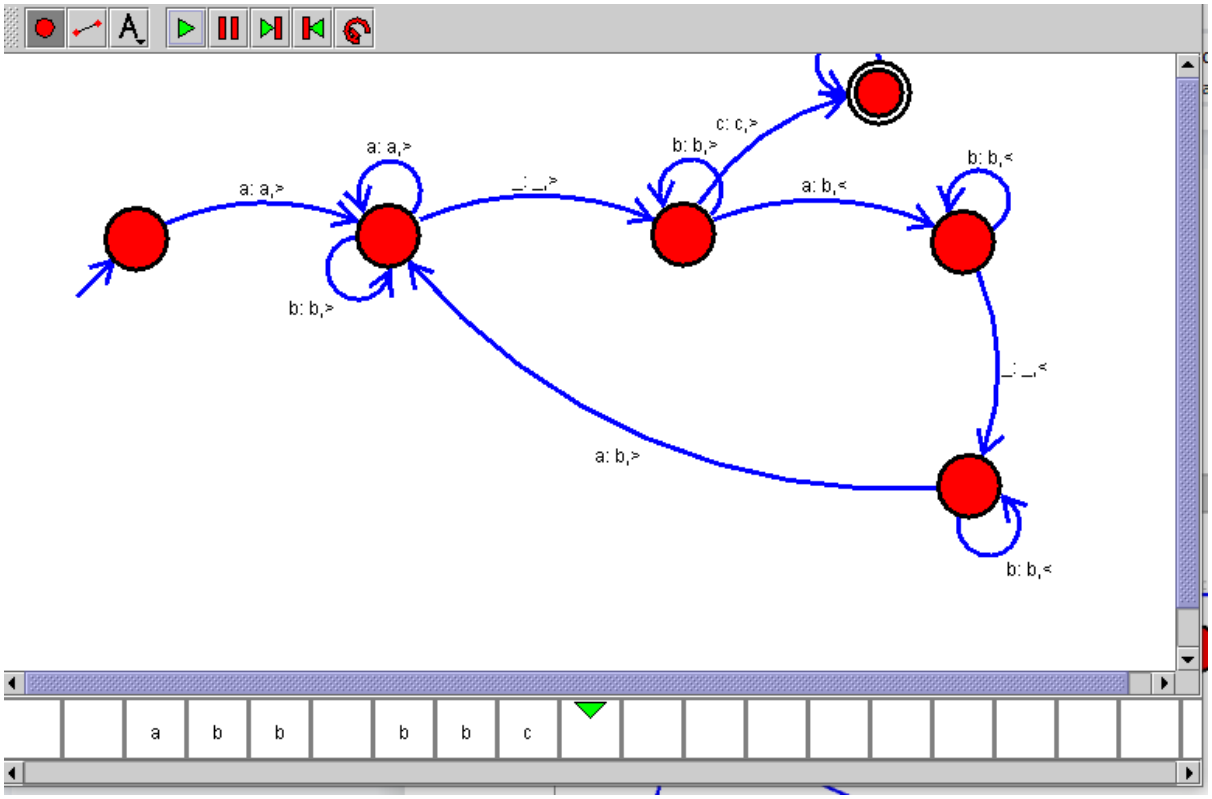
EXP :17

AIM: Design TM using simulator to perform subtraction of aaa-aa

Logic

W= aaa-aa

The Result of Subtraction is = a

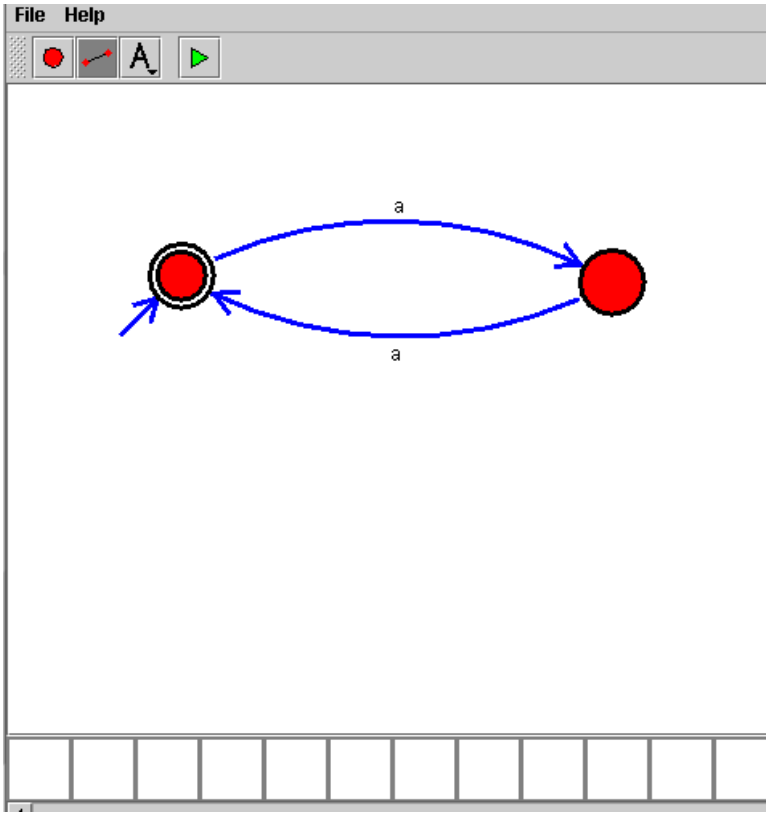


EXP :18

AIM: Design DFA using simulator to accept even number of a's

W{ aa, aaaa, aaaaaa}

Simulation

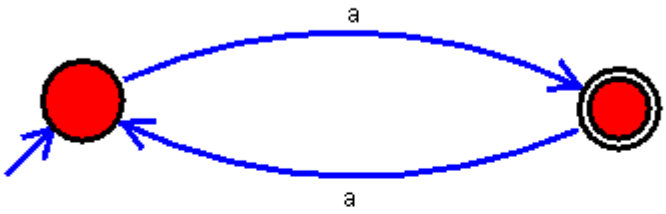


EXP :19

AIM: Design DFA using simulator to accept odd number of a's

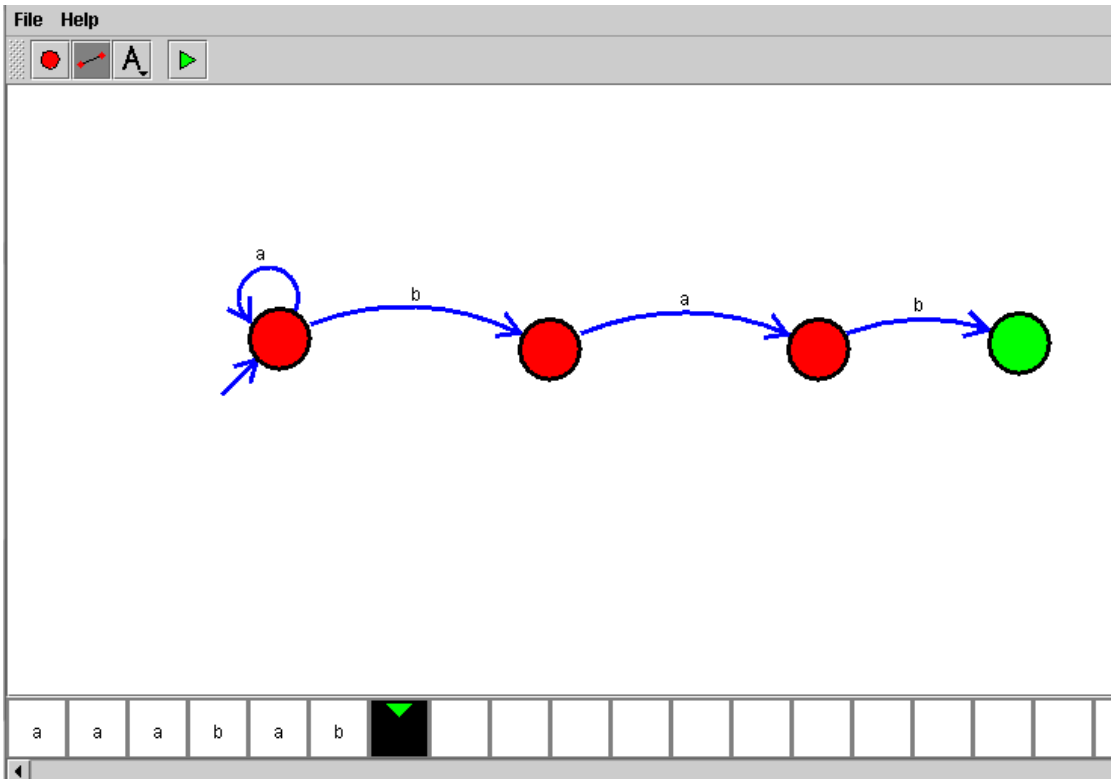
W{a, aaa, aaaaa}

Simulation



EXP :20

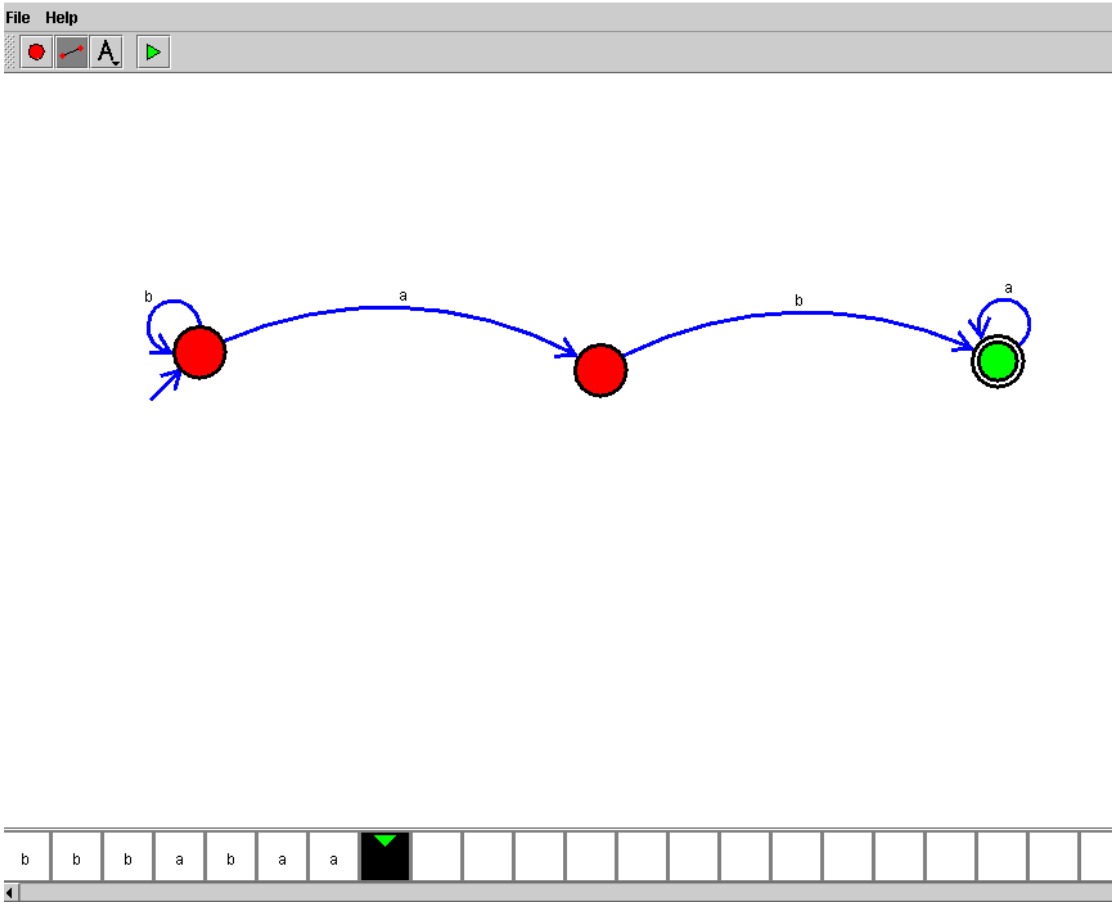
AIM: Design DFA using simulator to accept the string the end with ab over set {a,b) W= aaabab



EXP :21

AIM: Design DFA using simulator to accept the string having ‘ab’ as substring over the set {a,b}

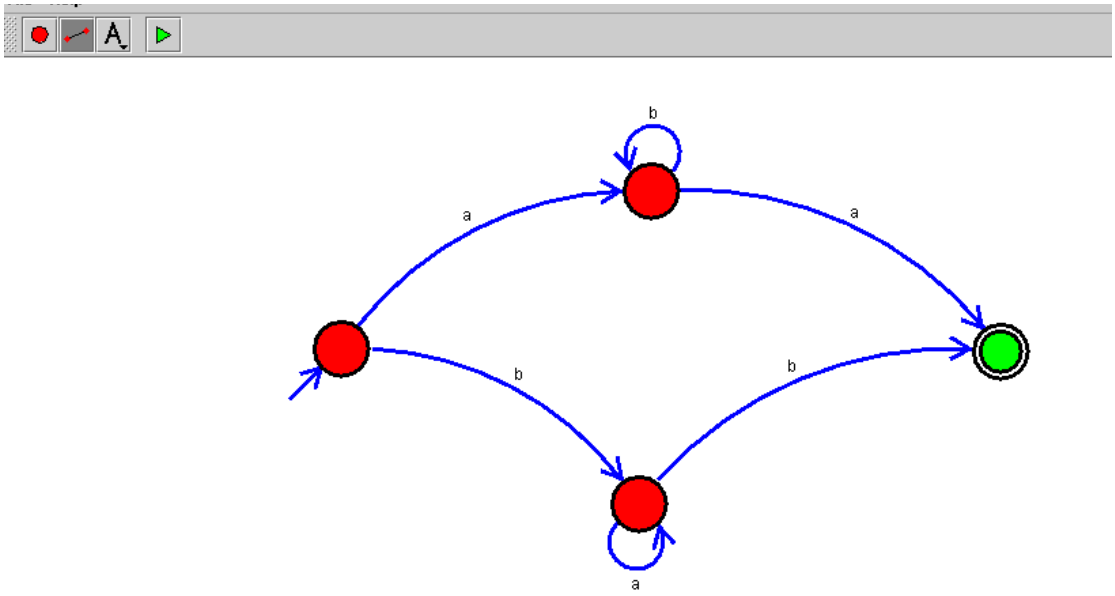
W= babaaaaa



EXP :22

AIM: Design DFA using simulator to accept the string start with a or b over the set {a,b}

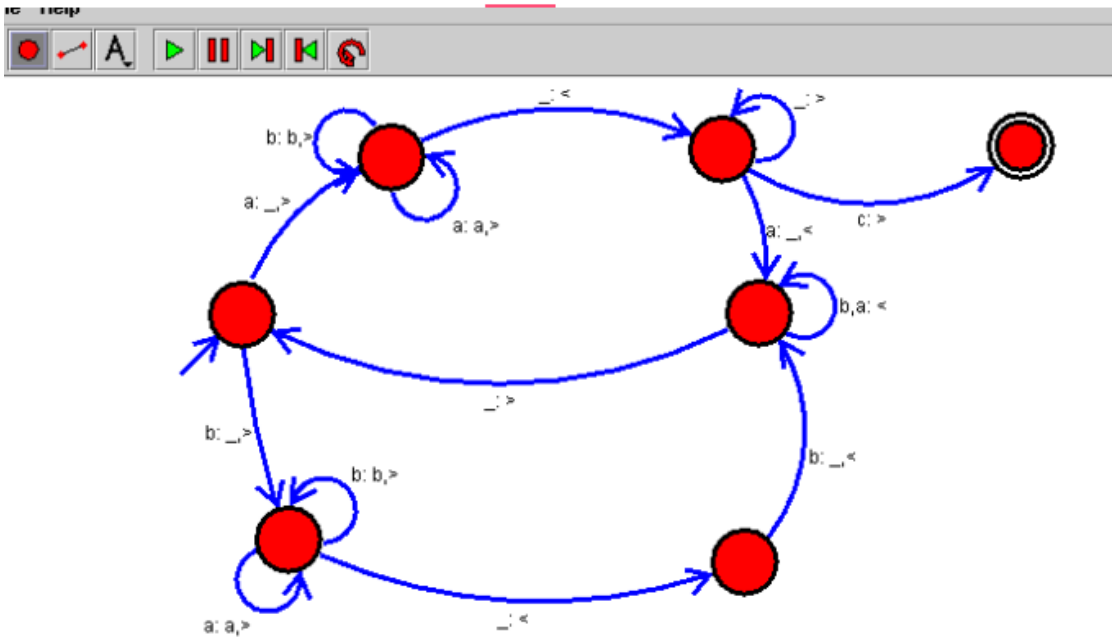
W= { abbbba, baaaaab}



EXP :23

AIM: Design TM using simulator to accept the input string Palindrome bbabb

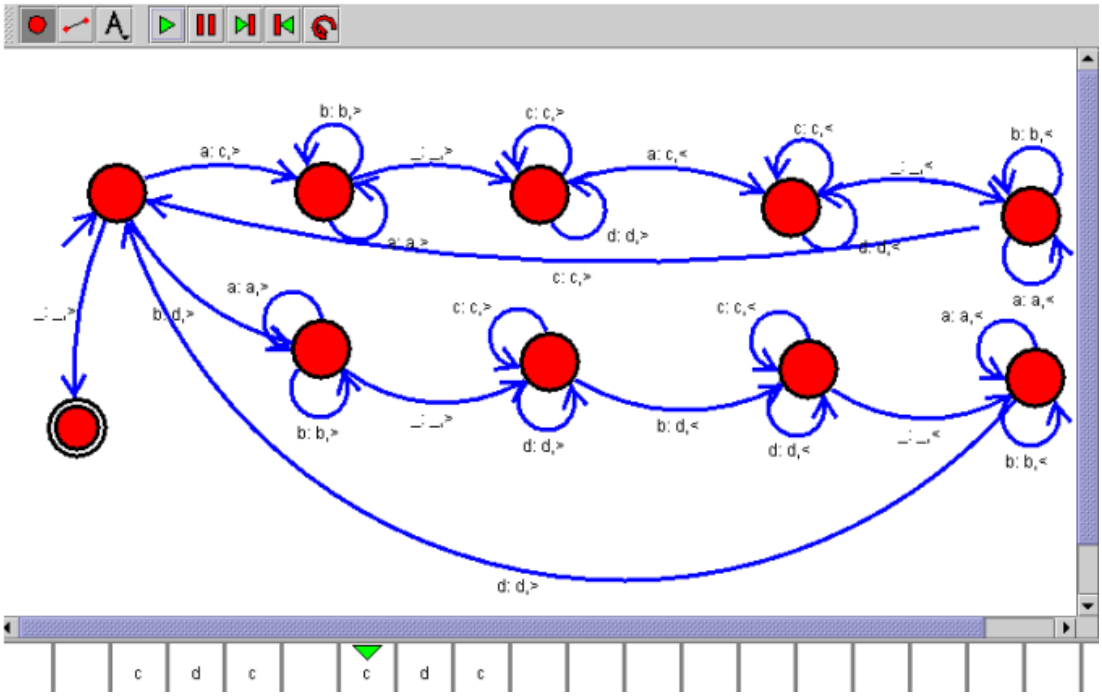
W={bbabb}



EXP :24

AIM: Design TM using simulator to accept the input string wcw

W={ aa aa,bb bb , ab ab}

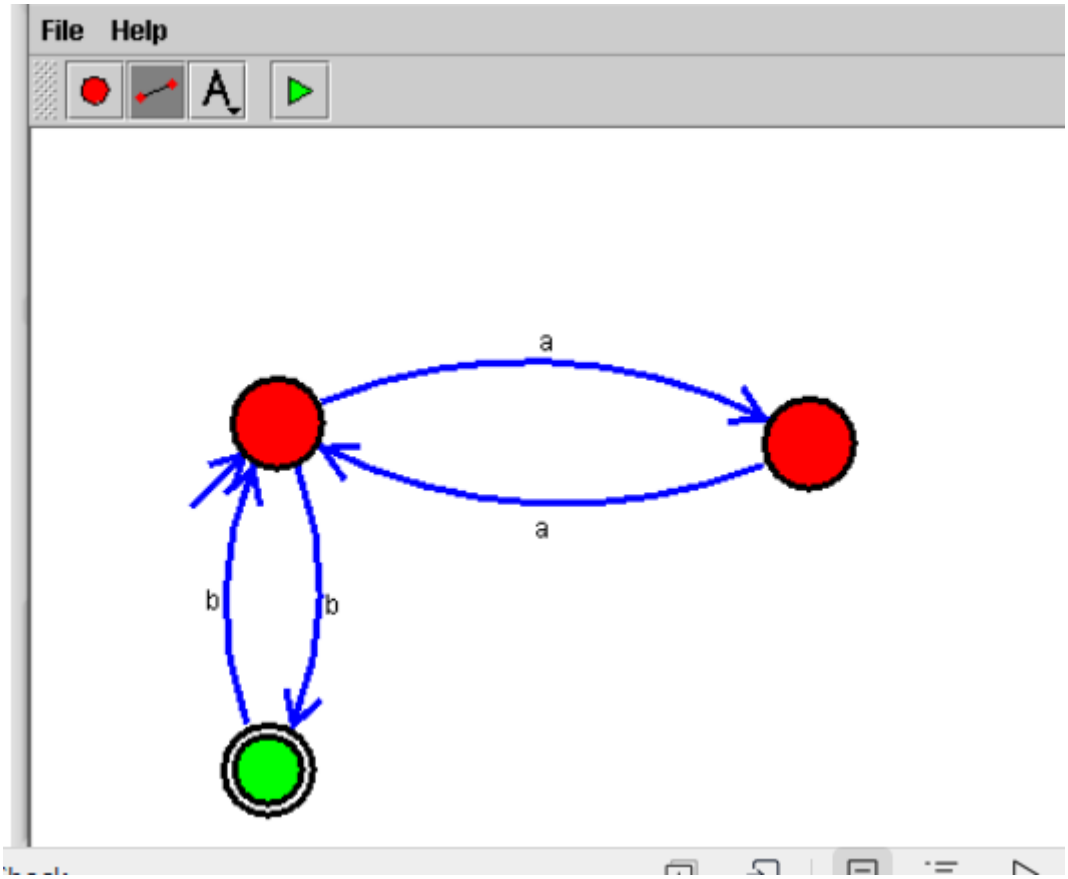


EXP :25

AIM: Design DFA using simulator to accept the string even number of a's and odd number of b's

W={aab, bbaab}

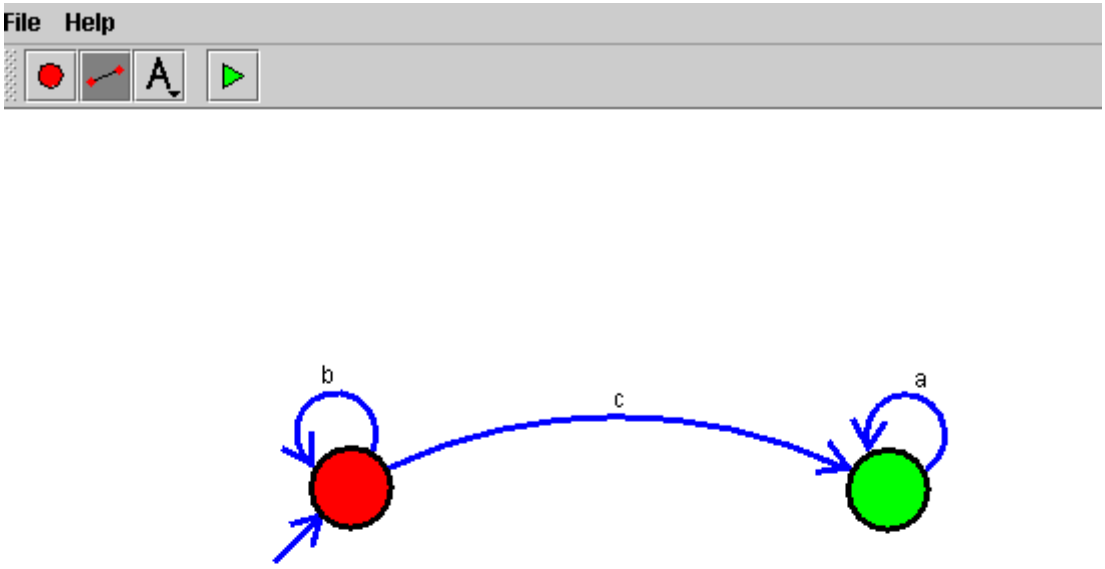
Simulation



EXP :26

AIM: Design DFA using simulator to accept the input string “bc” ,”c”,and ”bcaaa”

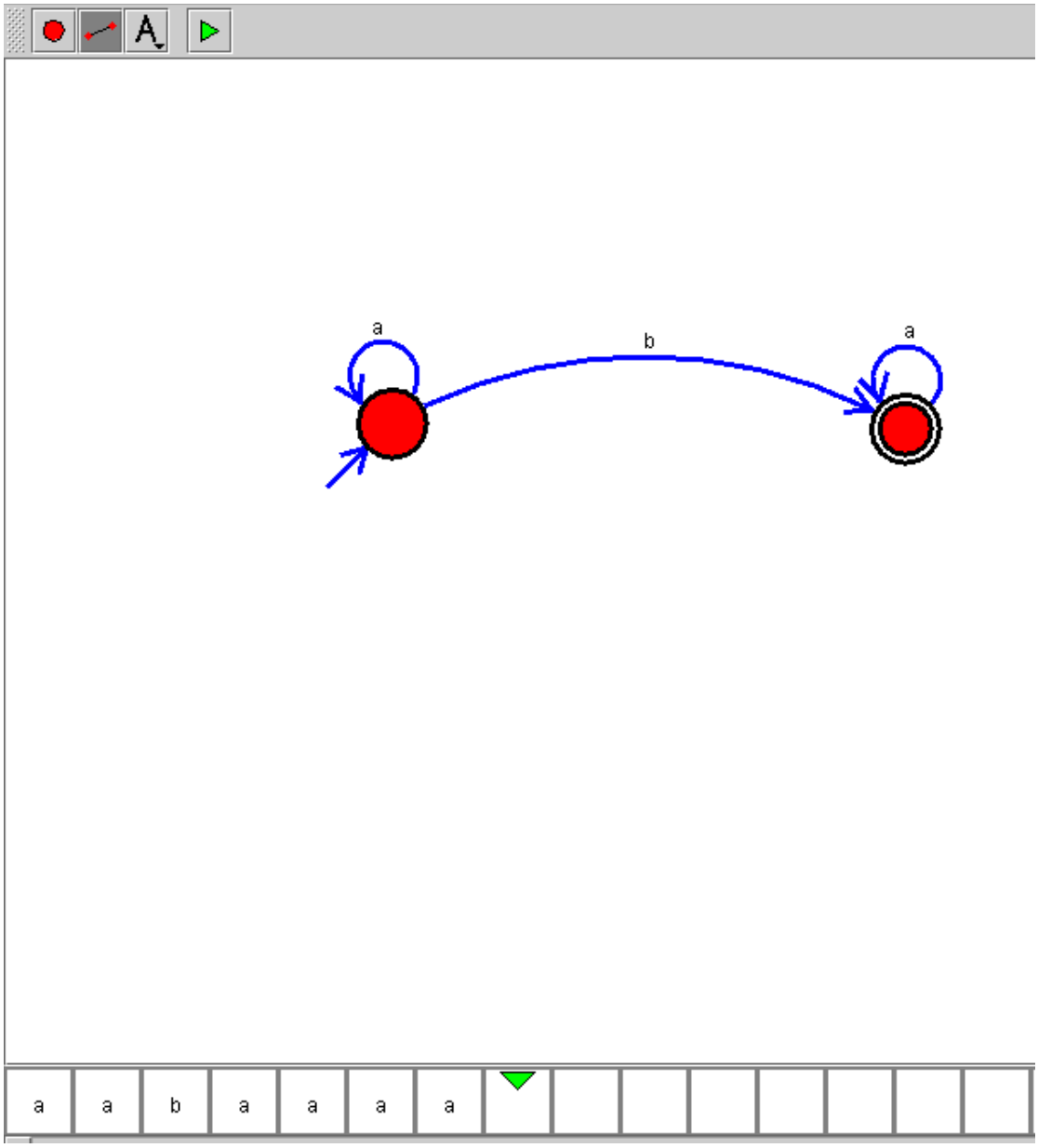
Simulation



EXP :27

AIM: Design NFA to accept any number of a's where input={a,b}

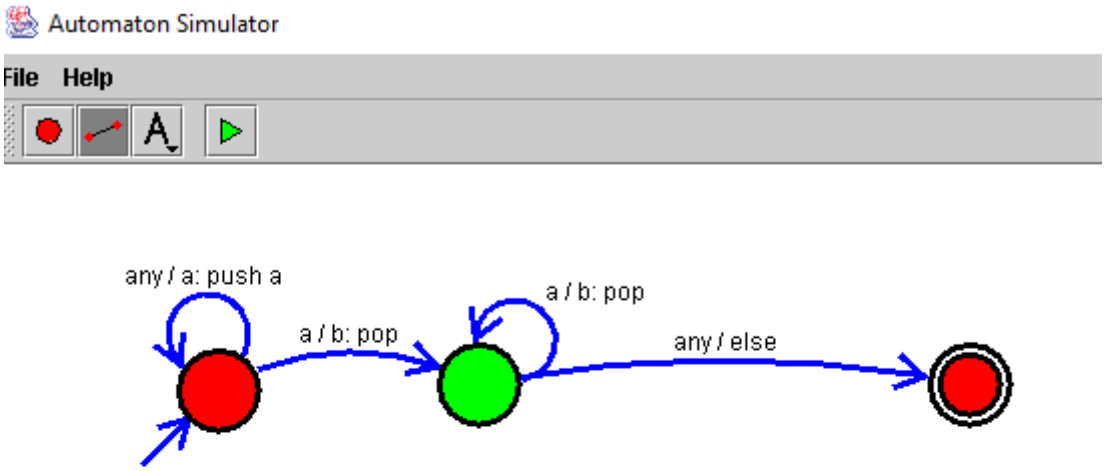
W={ aaaab, baaaaaa}



EXP :28

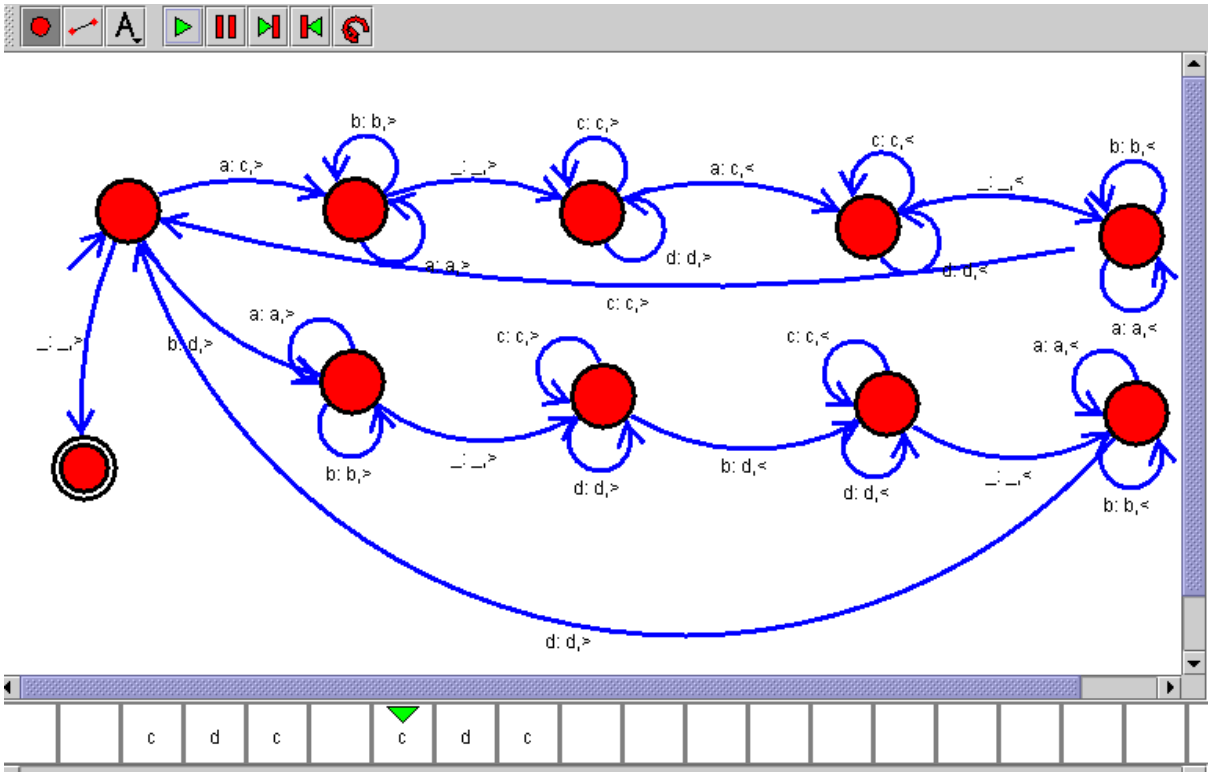
AIM: Design PDA using simulator to accept the input string $a^n b^n$

W={ aabb , aaabbb}



EXP :29

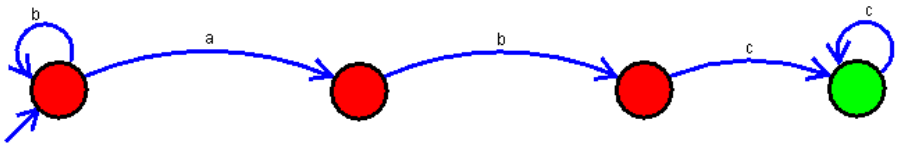
AIM: Design TM using simulator to perform string comparison where w={aba aba}



EXP :30

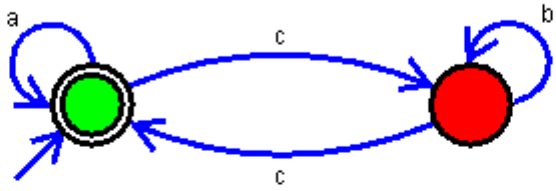
AIM: Design DFA using simulator to accept the string having ‘abc’ as substring over the set {a,b,c}

W= { aaaabcccc, abccccc}



EXP :31

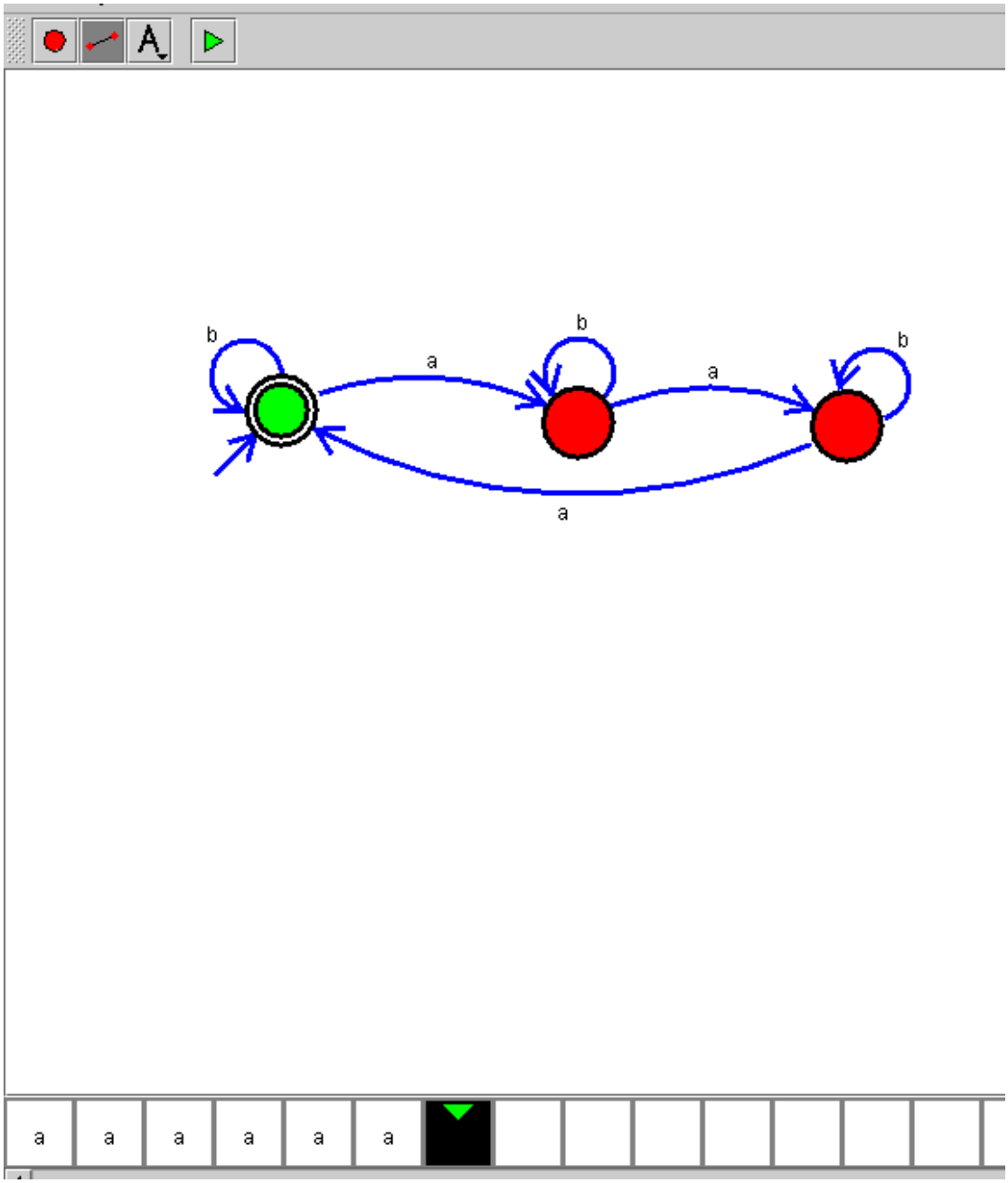
AIM: Design DFA using simulator to accept even number of c's over the set {a,b,c}



EXP :32

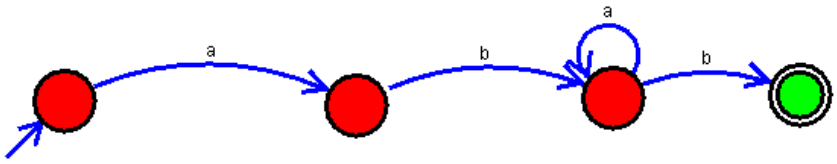
AIM: Design DFA using simulator to accept strings in which a's always appear tripled over input {a,b}

W={ aaa, ababa}



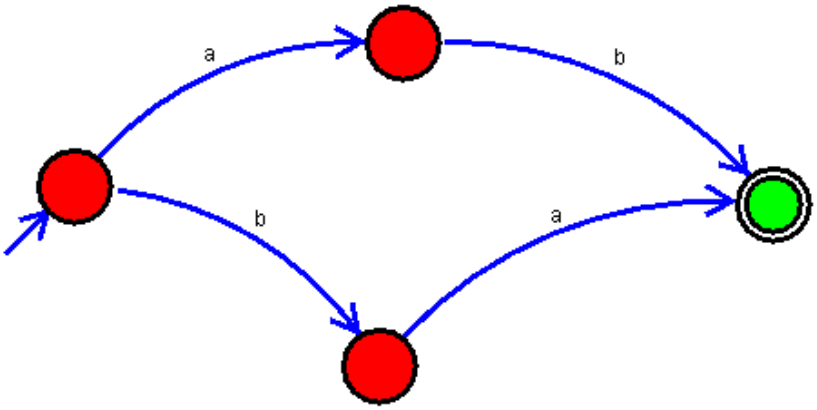
EXP :33

AIM: Design NFA using simulator to accept the string the start with a and end with b over set {a,b} and check W= abaab



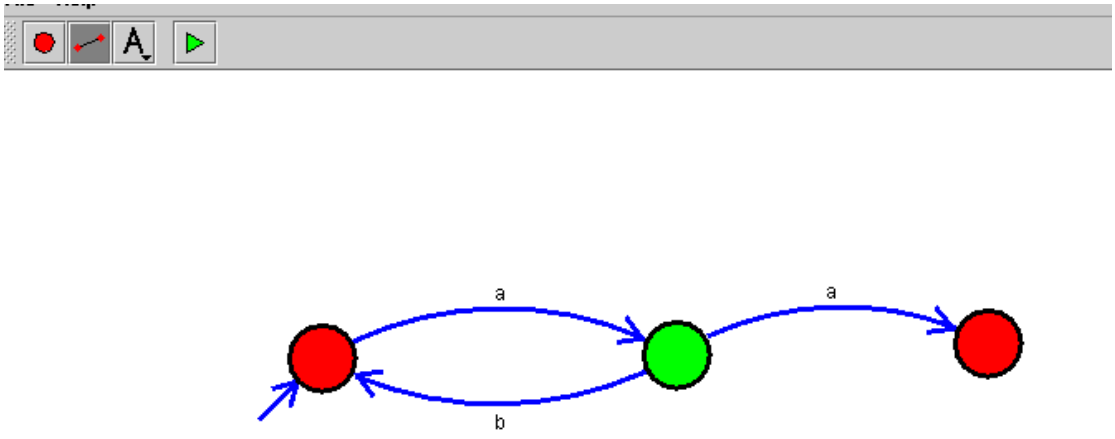
EXP :34

AIM: Design NFA using simulator to accept the string that start and end with different symbols over the input {a,b}



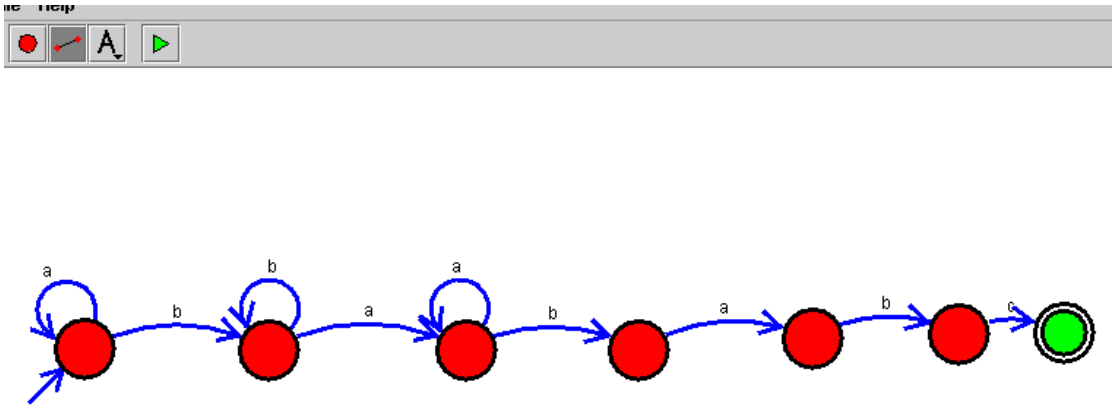
EXP :35

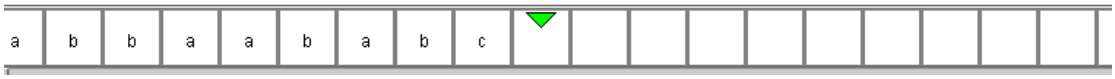
AIM: Let L be regular language, L consist set of string over { a,b) number a's minus number b's less than or equal to 2. Design DFA to accept the the language L.



EXP :36

AIM: Design DFA using simulator to accept the string the end with abc over set {a,b,c) W= abbaababc





EXP :37

AIM: Design NFA to accept any number of b's where input={a,b}