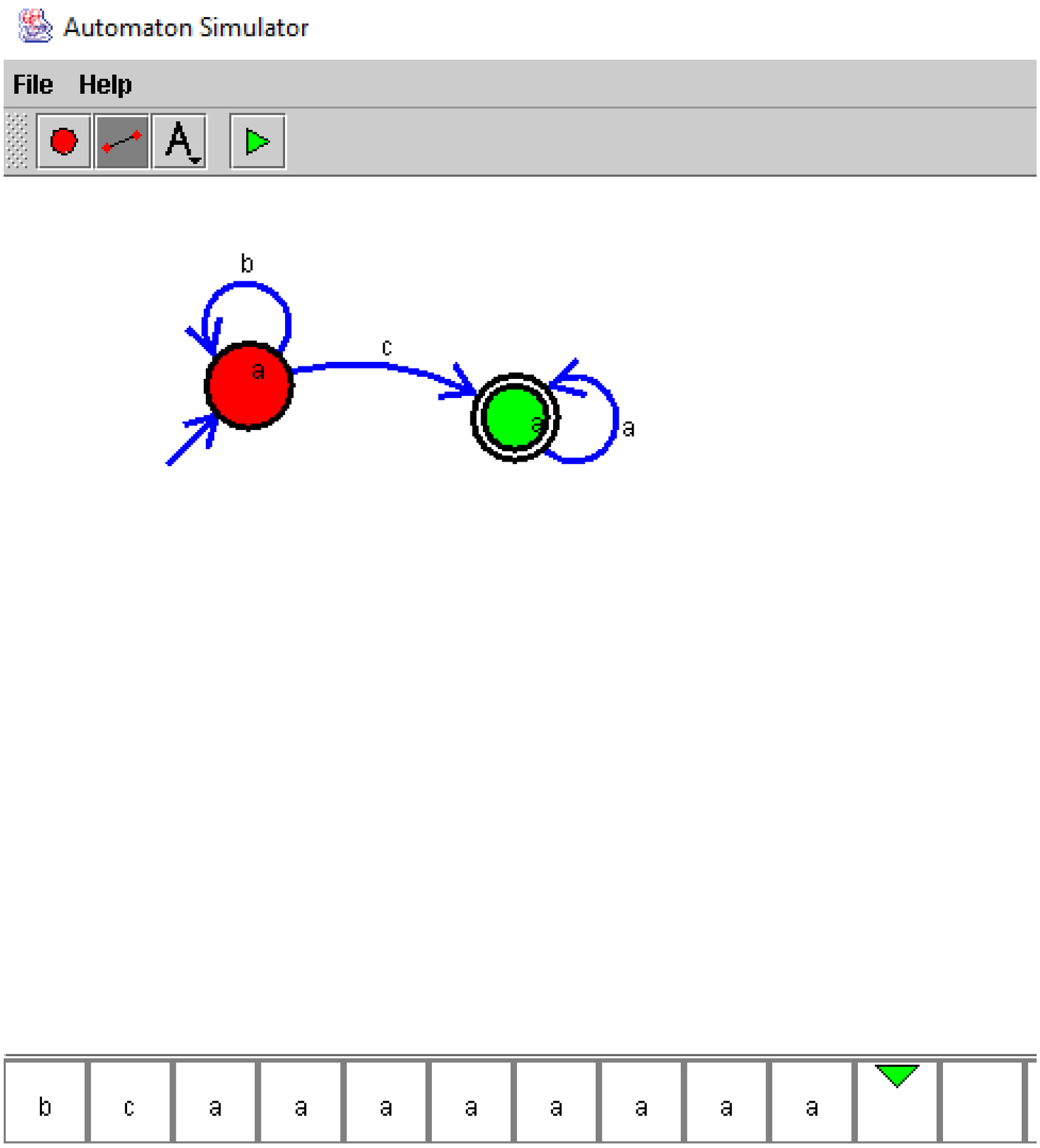
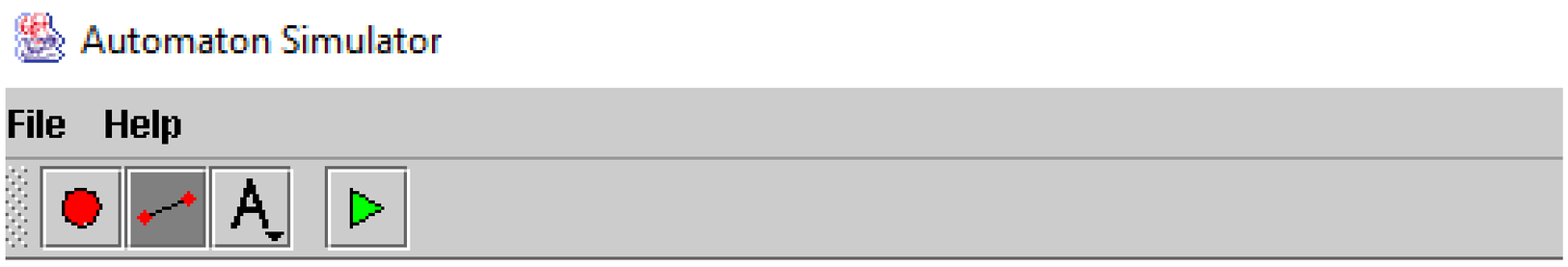
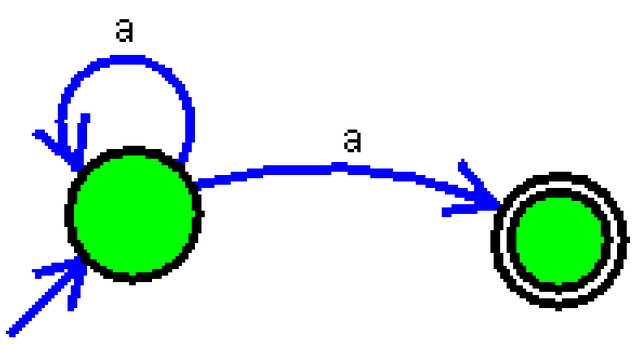
http://www.cburch.com/proj/autosim/download.html

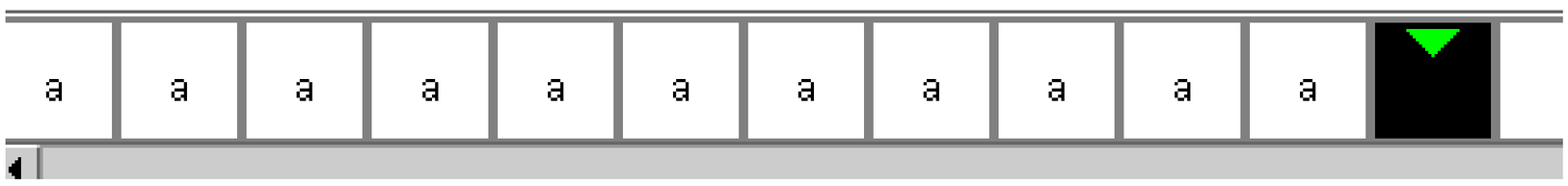
1. Design DFA to accept bcaaaaaaaaaaaaa, bc, and c



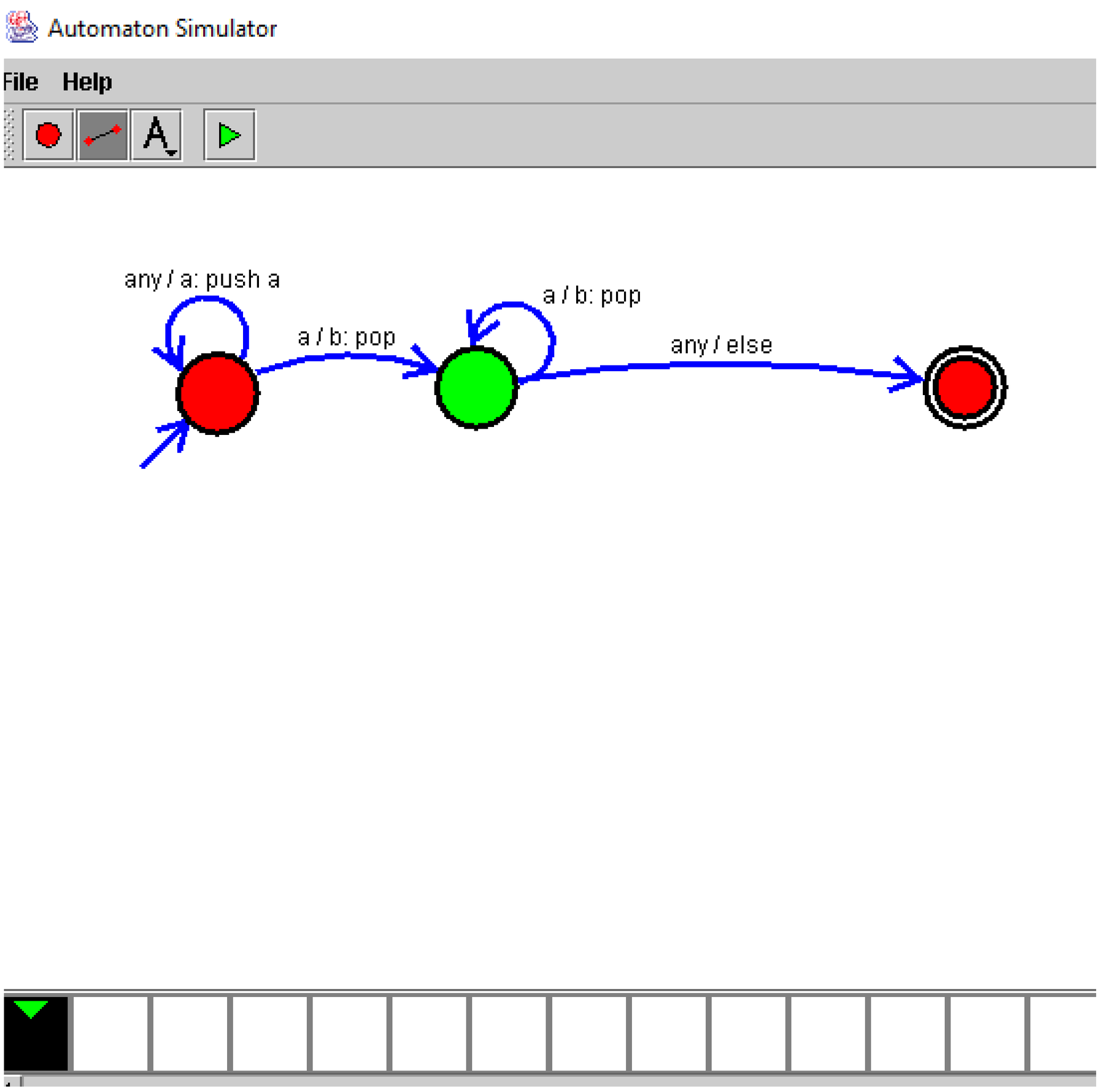
2.Design NFA to accept aaaaaa



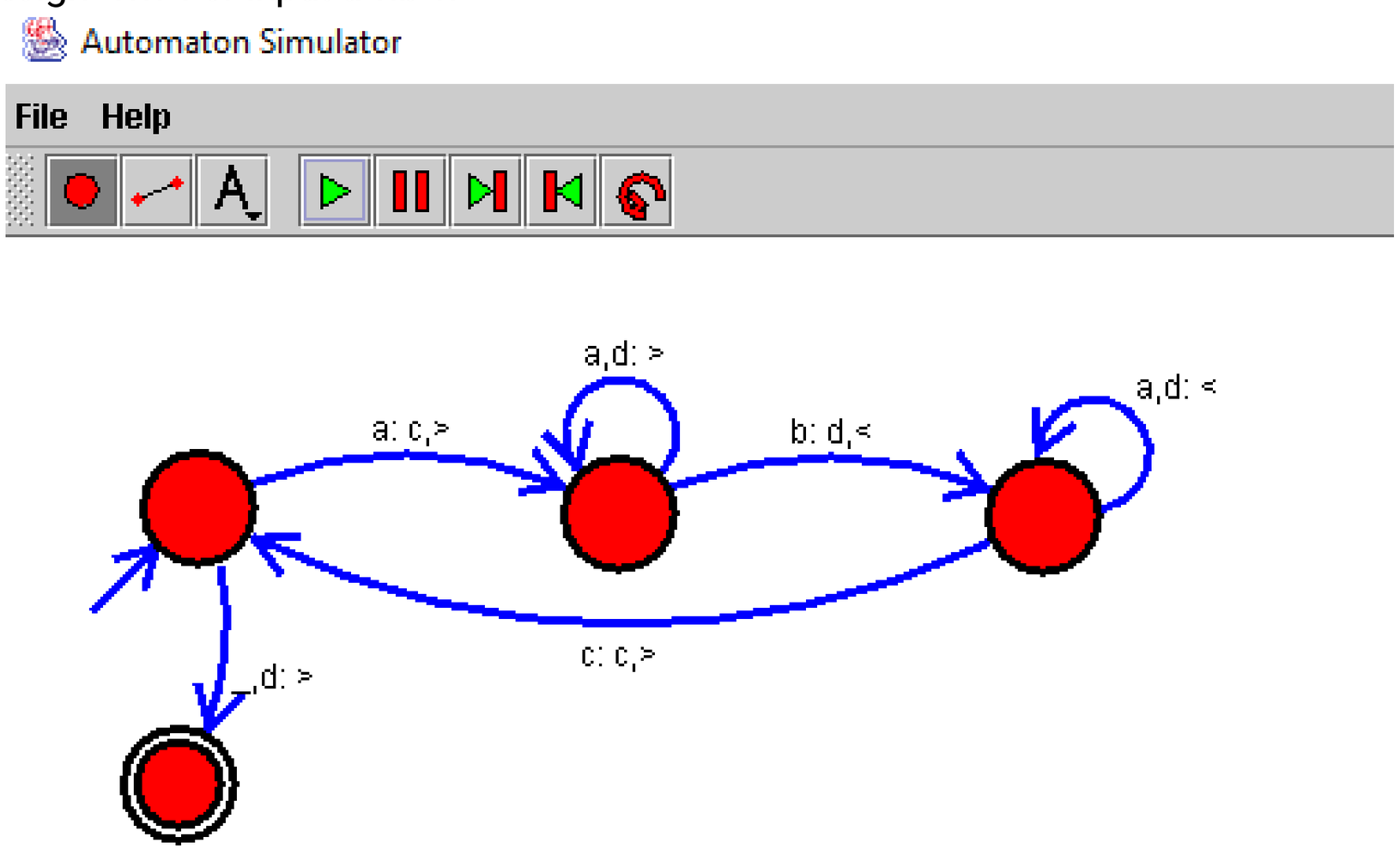




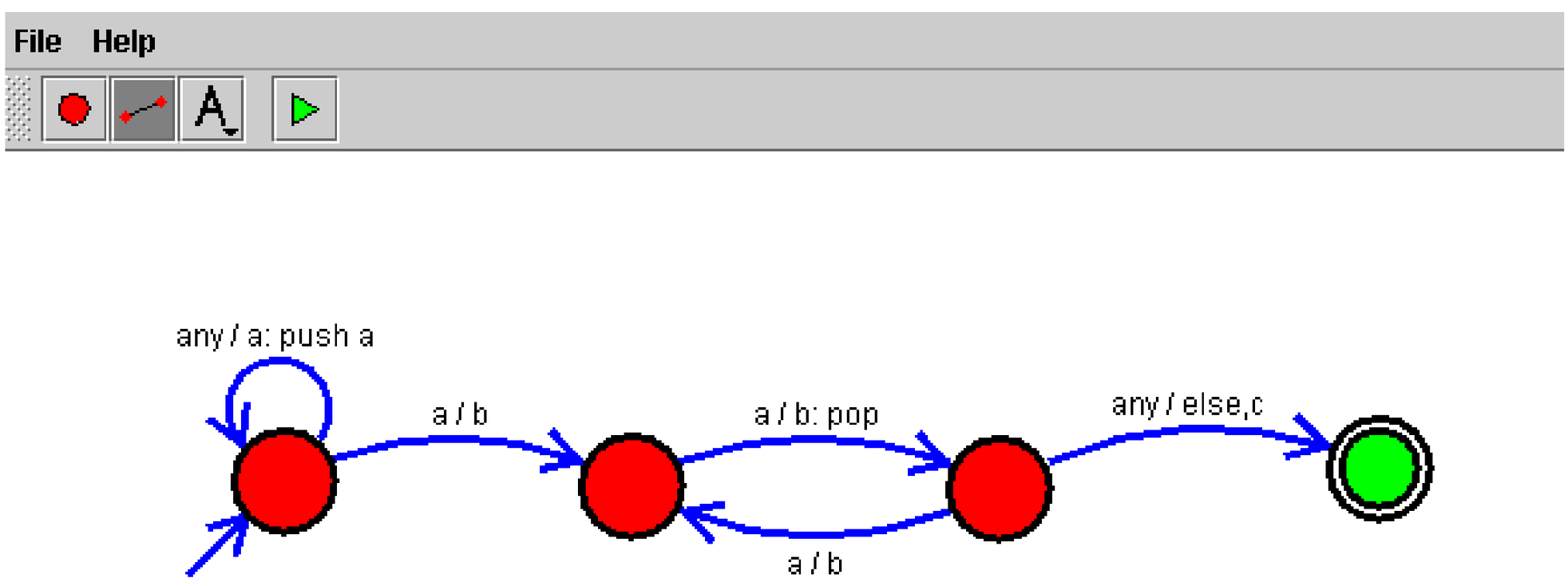
3.Design PDA for the input a^nb^n



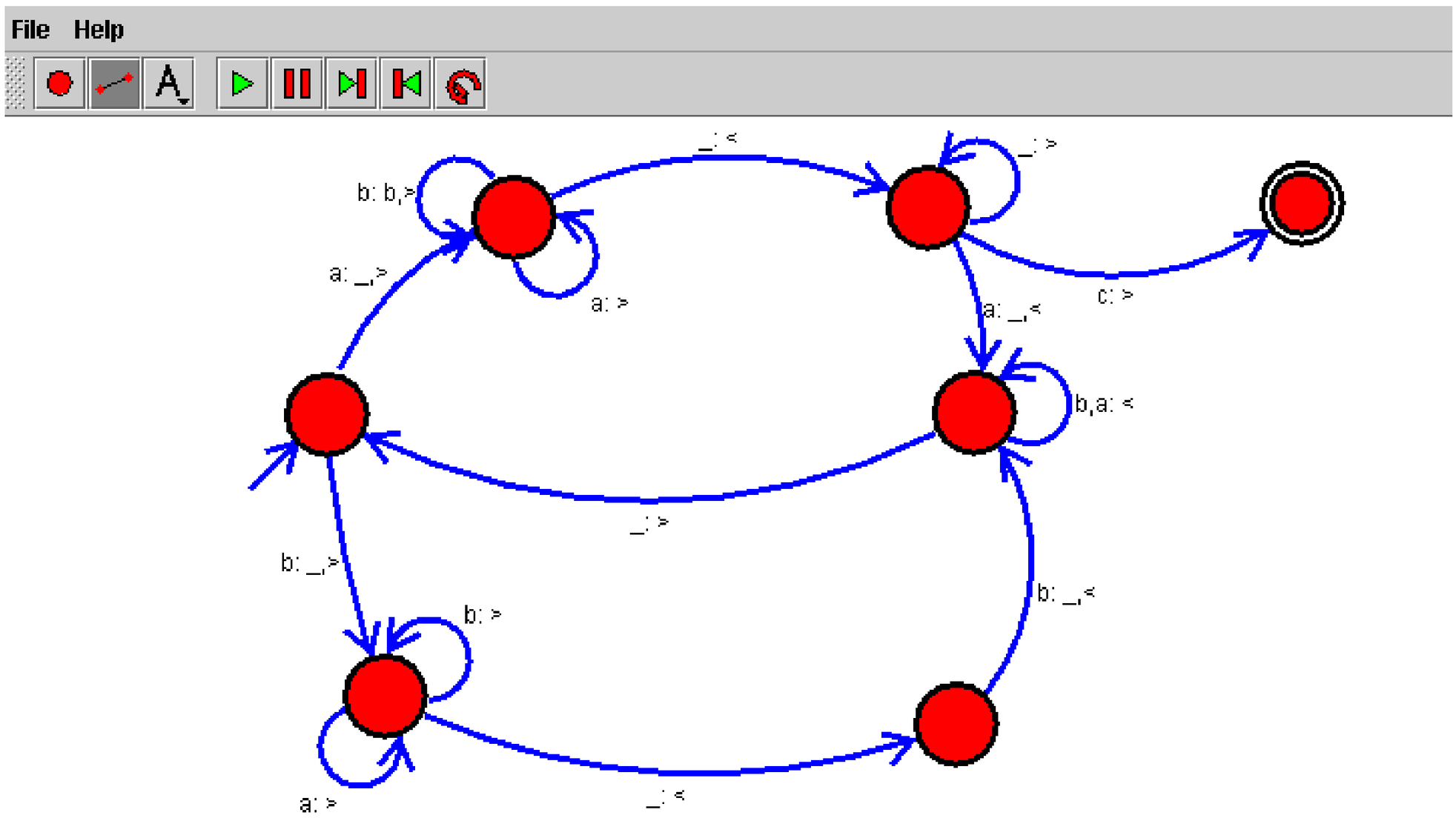
4.Design Tm For input a^nb^n



5 .Design PDA for input aabbbbc (L=a^nb^2n)



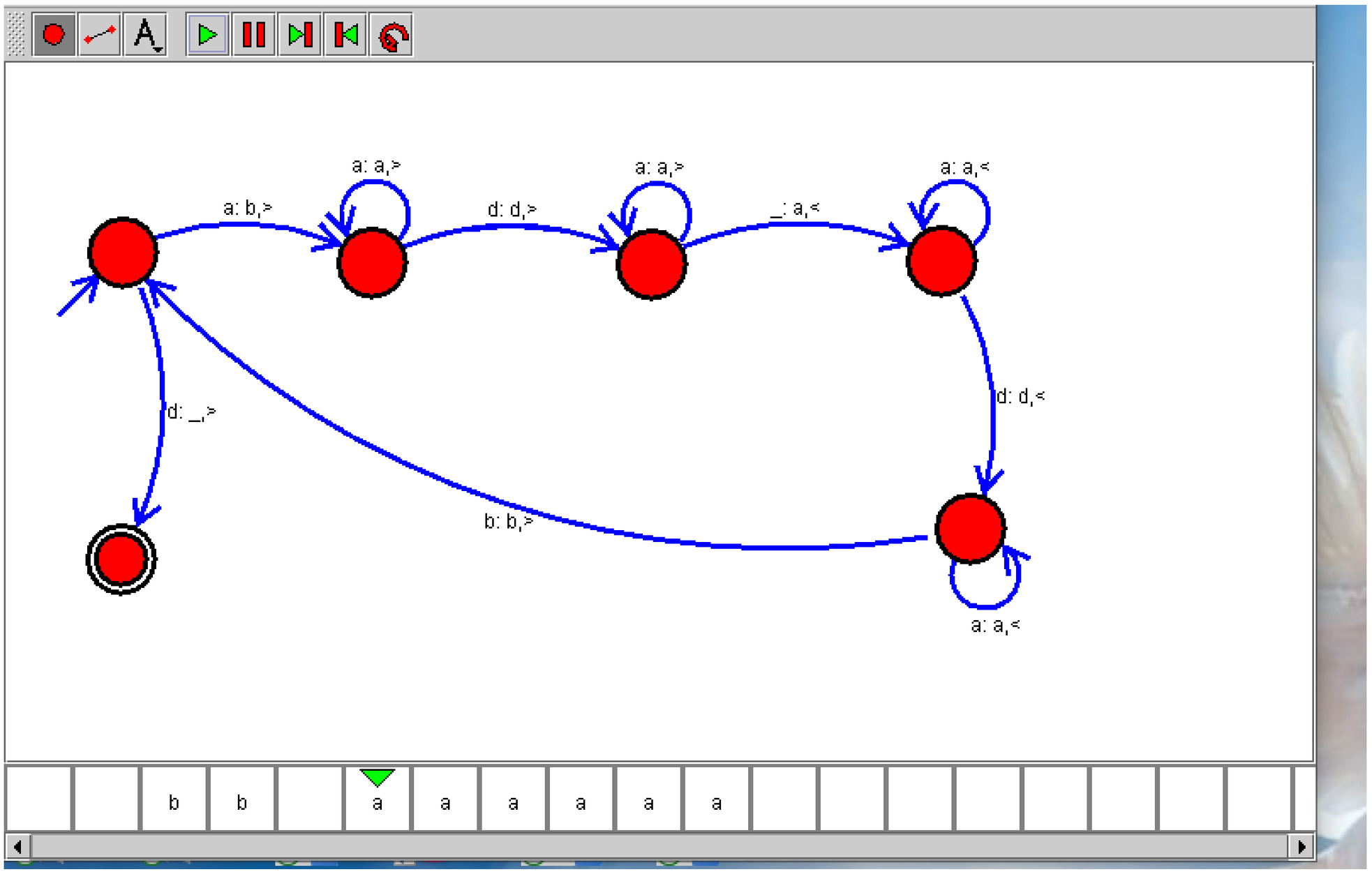
6.TM Simulation for Palindrome W= ababa c



7.Design TM to perform addition of following

W= aa + aaaa

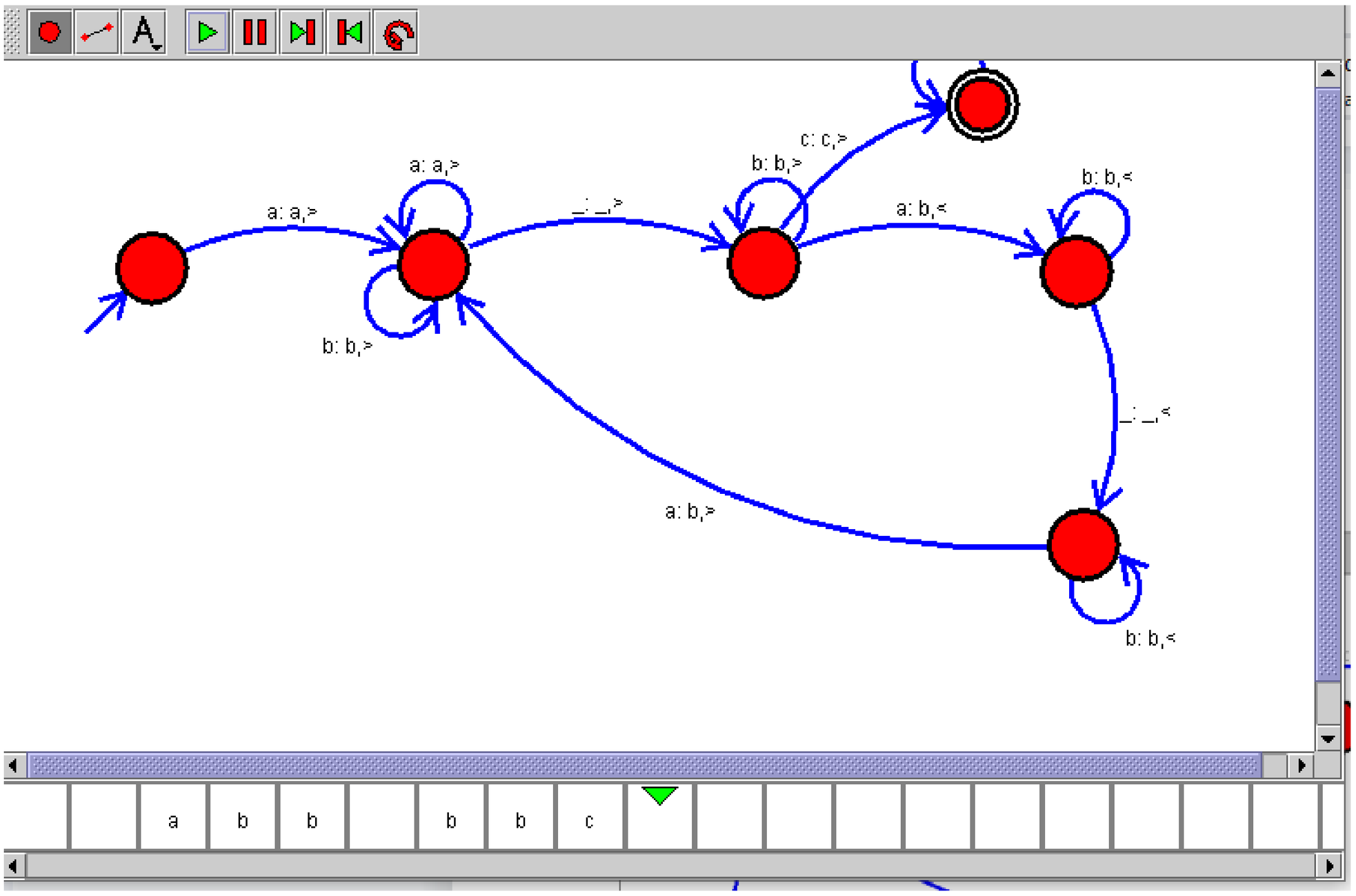
After Addition of a's = aaaaaa



8.Design TM to perform subtraction

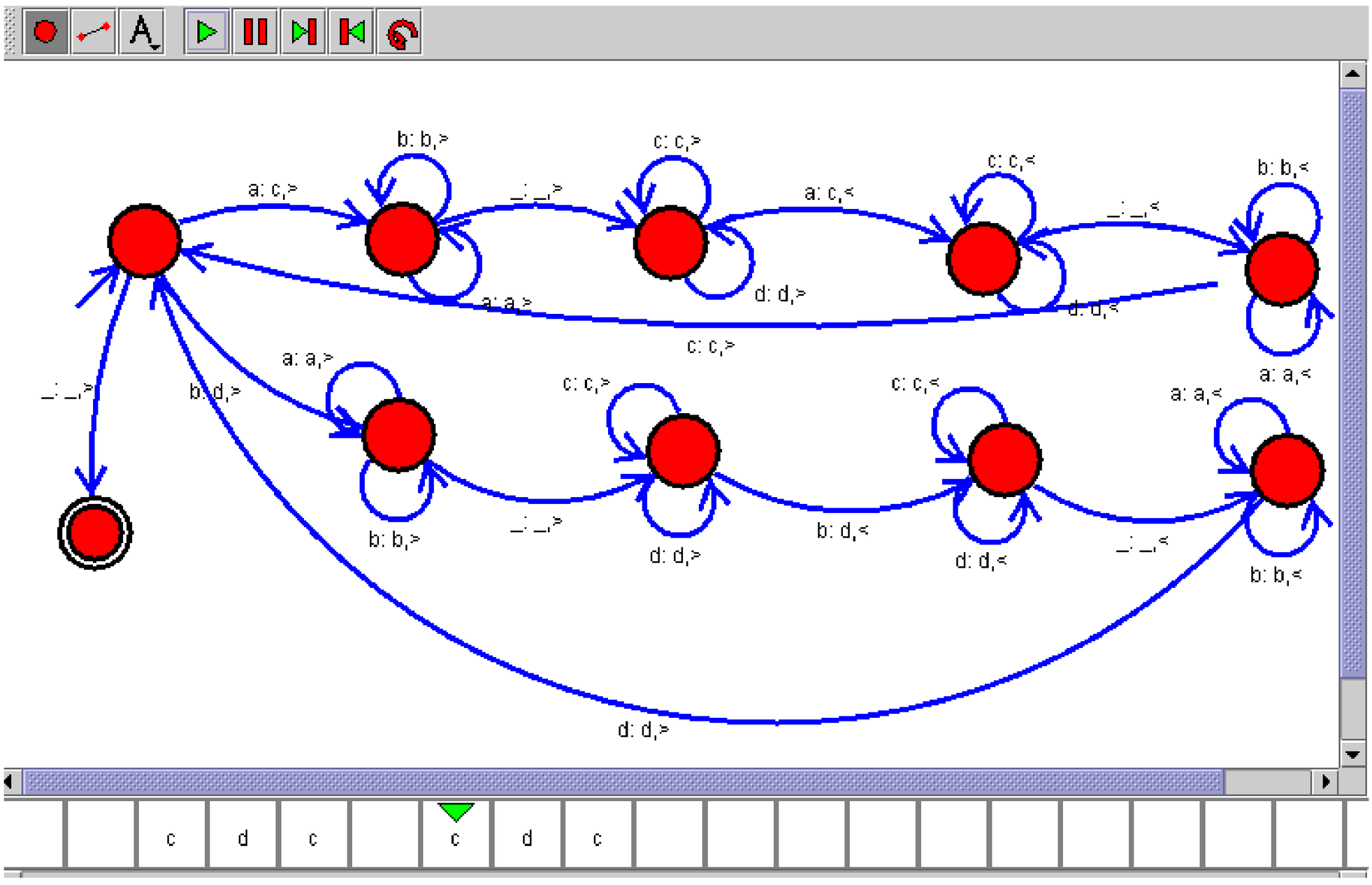
W= aaa-aa

The Result of Subtraction is = a



9.Design TM to perofrm string comparison

W = aba aba



https://web.stanford.edu/class/archive/cs/cs103/cs103.1156/tools/cfg/

10. Write CFG to product string which consists of substring ‘aa’

Create

Input your context-free grammar (CFG) here. The start symbol has already been filled in for you.

- The left-hand nonterminal of each production must be filled in.
- [ε] - An empty text field corresponds to epsilon.
- [|] - For "or", use the standard pipe character that you use while coding.
- Input is case-sensitive. Whitespace is not ignored.

Reset Example

S → AaaA

A → aA | bA | ε

Test

To test the CFG above, input test strings here, one per line. An empty line corresponds to the empty string. Results will be shown automatically.

aaab

Test Results for CFG

#	String	Matches	
1	"aaab"	Yes (ambiguously)	Derivation One Derivation Two

Rule	Application	Result
Start → **S**	Start	**S**
S → **A**aa**A**	**S**	**A**aa**A**
A → a**A**	**A**aa**A**	a**A**aa**A**
A → ε	a**A**aa**A**	aaa**A**
A → b**A**	aaa**A**	aaab**A**
A → ε	aaab**A**	aaab

11.Write Context Free Grammar to Generate Palindrome

Verify

This is the CFG you have input above:

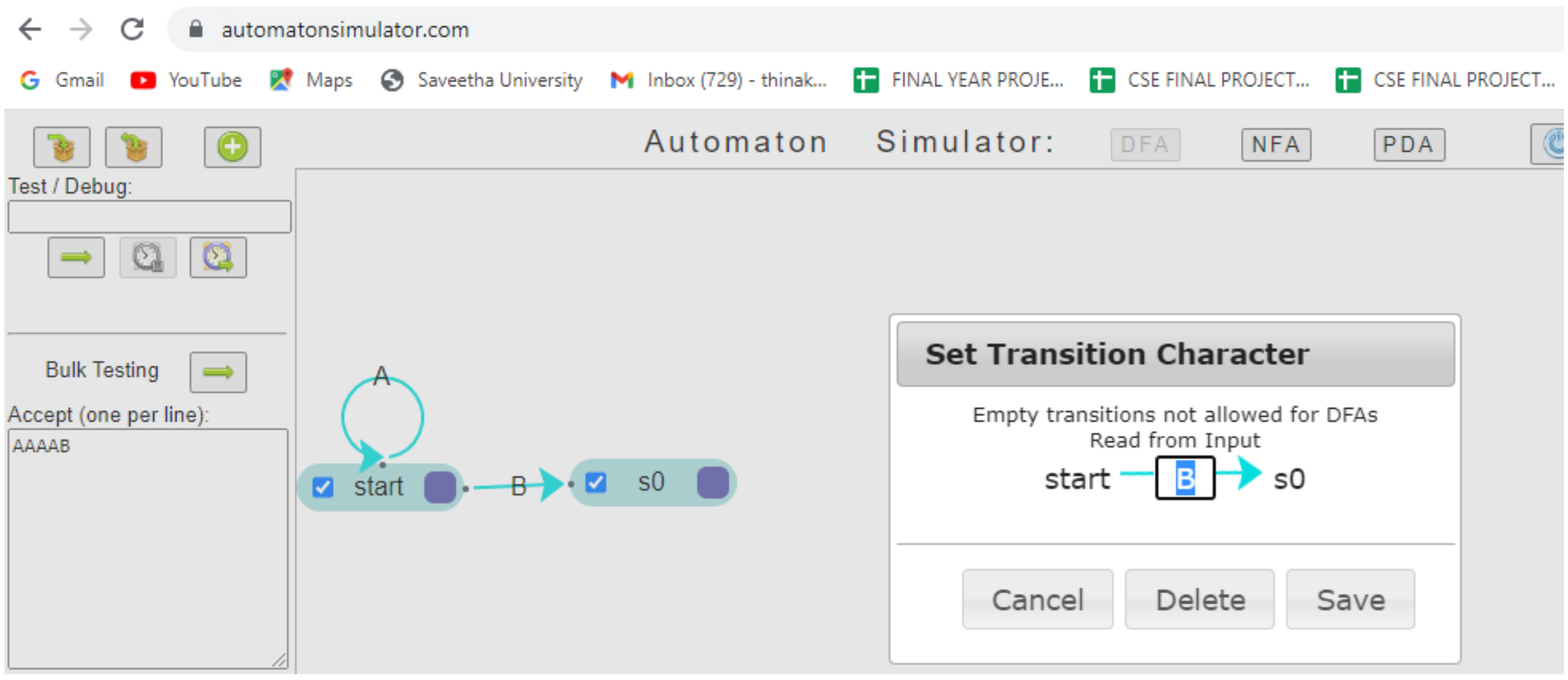
Start symbol: **S**

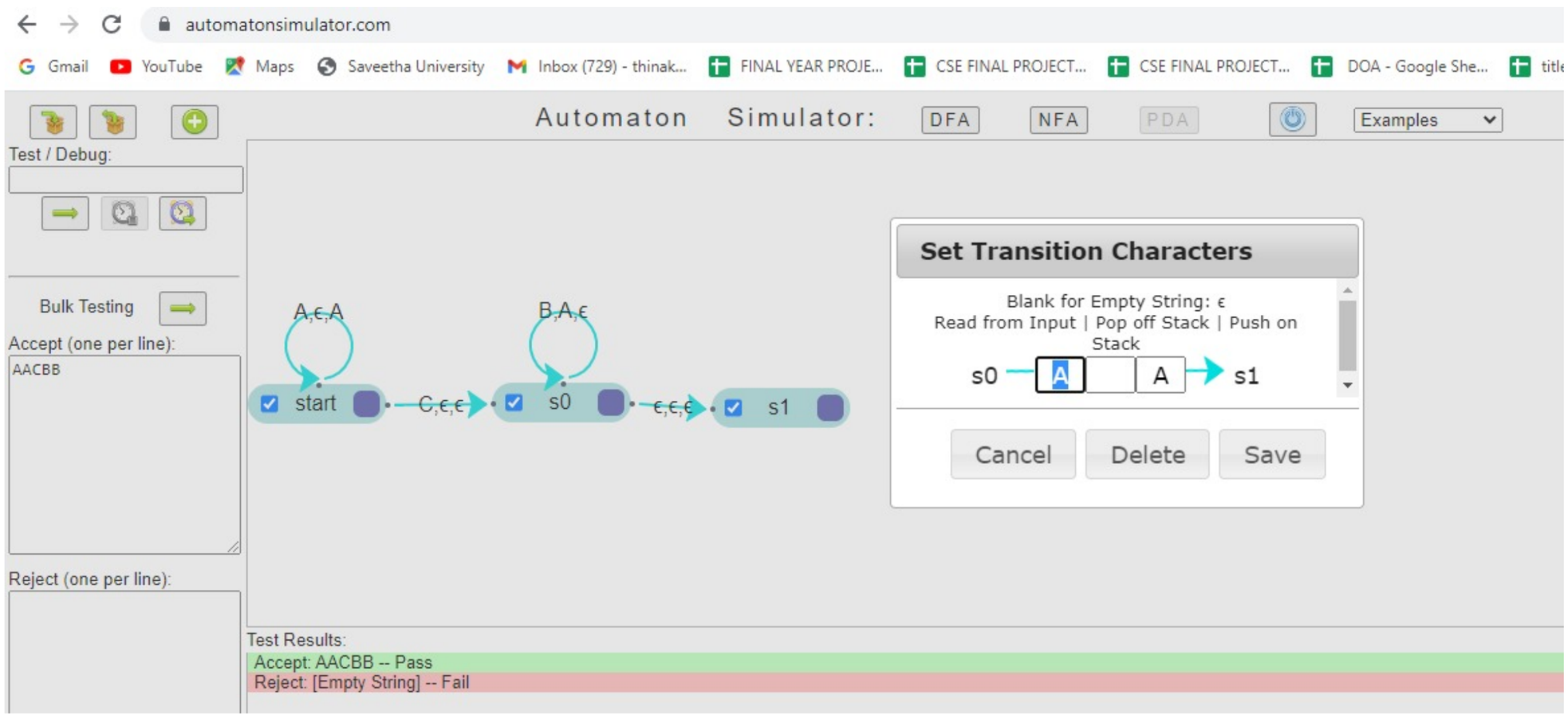
S → ε | a**S**a | b**S**b | a | b

Some strings from the language of this grammar:

```
abba
bbbbb
ababababbbbbababa
```

12.


automatonsimulator.com
Gmail
YouTube
Maps
Saveetha University
Inbox (729) - thinak...
FINAL YEAR PROJE...
CSE FINAL PROJECT...
CSE FINAL PROJECT...
Automaton Simulator:
DFA
NFA
PDA
Test / Debug:
Bulk Testing
Accept (one per line):
AAAAB
A
start
B
s0
Set Transition Character
Empty transitions not allowed for DFAs
Read from Input
start
B
s0
Cancel
Delete
Save


automatonsimulator.com
Gmail
YouTube
Maps
Saveetha University
Inbox (729) - thinak...
FINAL YEAR PROJE...
CSE FINAL PROJECT...
CSE FINAL PROJECT...
DOA - Google She...
Automaton Simulator:
DFA
NFA
PDA
Examples
Test / Debug:
Bulk Testing
Accept (one per line):
AACBB
Reject (one per line):
A,ϵ,A
B,A,ϵ
start
C,ϵ,ϵ
s0
ϵ,ϵ,ϵ
s1
Set Transition Characters
Blank for Empty String: ϵ
Read from Input | Pop off Stack | Push on Stack
s0
A
A
s1
Cancel
Delete
Save
Test Results:
Accept: AACBB -- Pass
Reject: [Empty String] -- Fail

13.

Automaton Simulator:

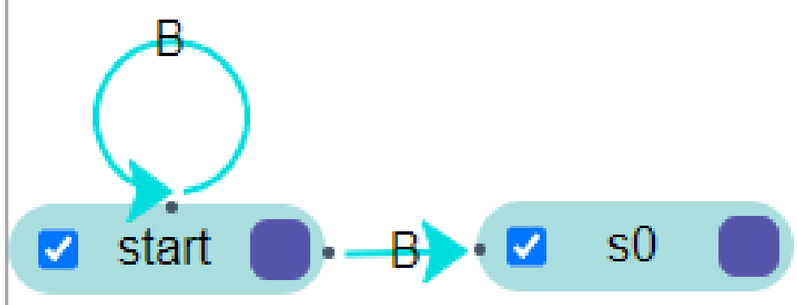
est / Debug:

Bulk Testing

ccept (one per line):

BBBBBBBBBBB

eject (one per line):



Test Results:

Accept: BBBBBBBBBBB -- Pass

14. Write CFG which will produce string over set = {a,b} that start with ‘a’ and end with ‘b’

Verify

This is the CFG you have input above:

Start symbol: **S**

S → ε | a**A**b

A → a**A** | b**A** | ε

Some strings from the language of this grammar:

```
aabbaab
abb
aaabaabbaababab
abbb
aabaaaaab
abaababb
abaabbaab
ε
abbbab
aabbababbabbb
```

15. Design PDA to accept event number a's and even number b's