Ref: https://youtu.be/YDHn_SxEXV4?si=g5oF3XwGaQ3FnoRF

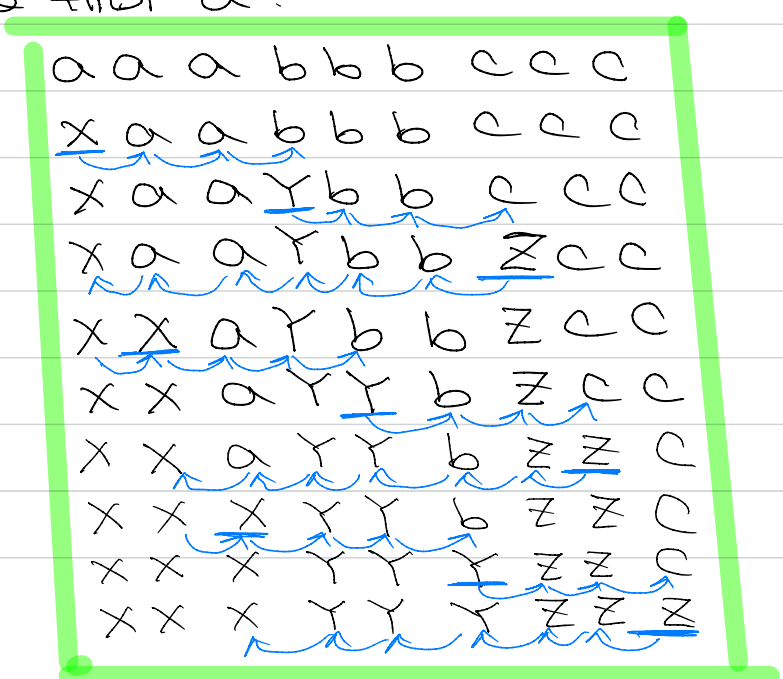
# Turing machine for $a^n b^n c^n$

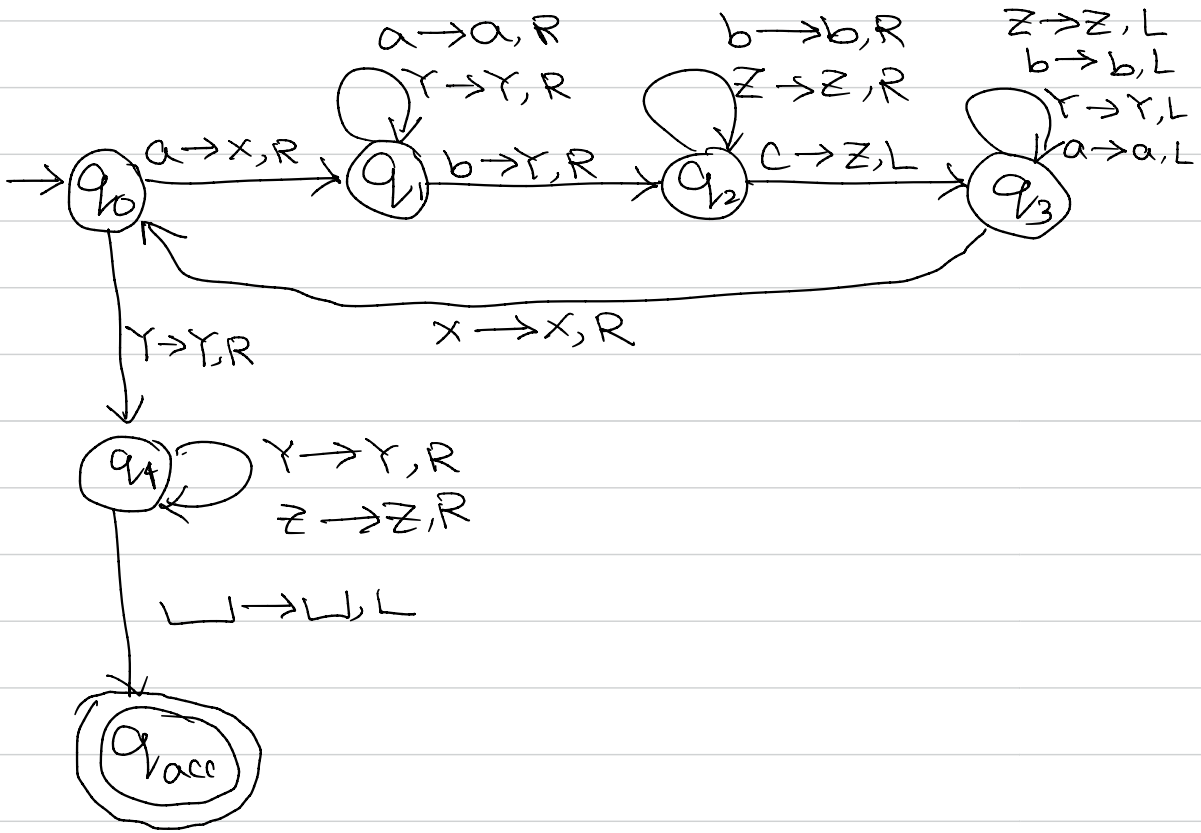
Algorithm

I) Replace first a with X & move right until first b appears.

II) Replace the first b with Y & move right until first c appears.

III) Replace the first c with Z & move left until the first a appears.
For this we move left until we get X & then move right to get the first a.




a → a, R
Y → Y, R
b → b, R
Z → Z, R
Z → Z, L
b → b, L
Y → Y, L
a → a, L
a → X, R
b → Y, R
C → Z, L
q0
q1
q2
q3
X → X, R
Y → Y, R
q4
Y → Y, R
Z → Z, R
␣ → ␣, L
qacc

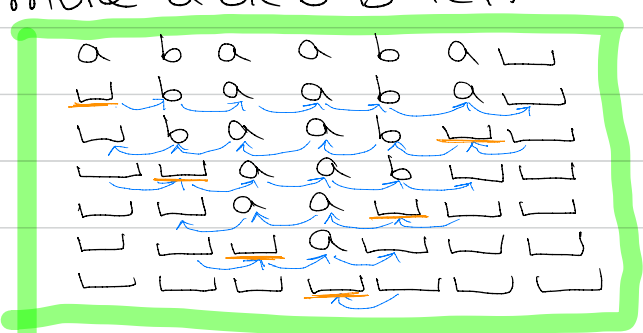
Ref: https://youtu.be/KW9md3j4_cU?si=0Dzm3zblXF2ZkQ29

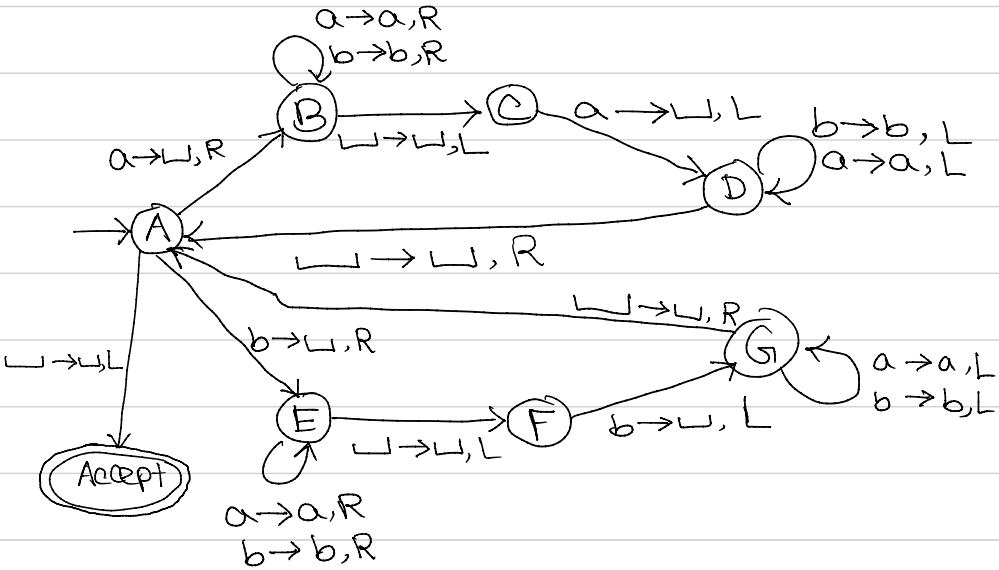
# Turing machine for even palindrome $wcw^R$

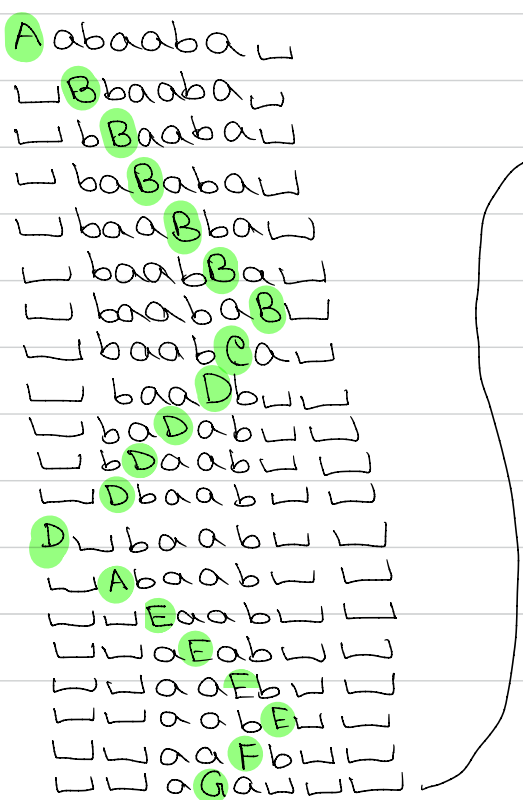
| a | b | a | a | b | a | ␣ | ... |
|---|---|---|---|---|---|---|---|

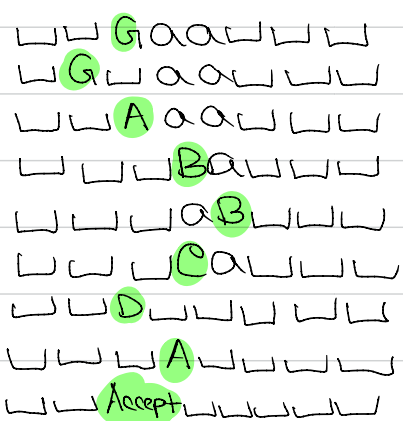
Algorithm:

1) Replace the first a or b with ␣ & move right to the end of the string.
2) When ␣ symbol is reached, it means we are at the end of the string. So move left. If the last character of the string is same as the character we replaced with ␣ in step 1 then replace the last character with ␣ & move left.
3) Move left until the first character a or b is reached & continue the process until no more a or b is left.




a→a,R
b→b,R
B
C
a→␣,L
b→b,L
a→a,L
D
a→␣,R
␣→␣,L
A
␣→␣,R
␣→␣,R
G
a→a,L
b→b,L
b→␣,R
␣→␣,L
E
F
b→␣,L
␣→␣,L
Accept
a→a,R
b→b,R


Aabaaba␣
␣Bbaaba␣
␣bBaaba␣
␣baBaba␣
␣baaBba␣
␣baabBa␣
␣baabaB␣
␣baabCa␣
␣baaDb␣␣
␣baDab␣␣
␣bDaab␣␣
␣Dbaab␣␣
D␣baab␣␣
␣Abaab␣␣
␣␣Eaab␣␣
␣␣aEab␣␣
␣␣aaEb␣␣
␣␣aabE␣␣
␣␣aaFb␣␣
␣␣aGa␣␣␣


␣␣Gaa␣␣␣
␣G␣aa␣␣␣
␣␣Aaa␣␣␣
␣␣␣Ba␣␣␣
␣␣␣aB␣␣␣
␣␣␣Ca␣␣␣
␣␣D␣␣␣␣
␣␣␣A␣␣␣␣
␣␣Accept␣␣␣␣

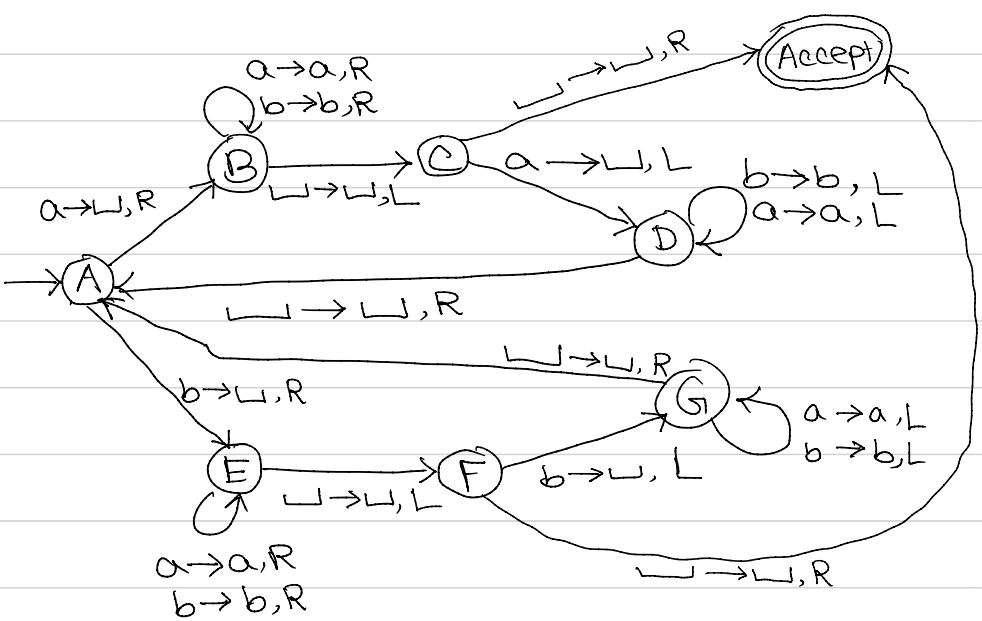
# Turing Machine for odd palindrom

$waw^R$ or $wbw^R$ where $w \in \{a, b\}$

Similar to even palindrom with some exceptions :-

1) for abaaaba, when the tape is ␣␣␣a␣␣␣ then replace a with ␣ & goto final state.

2) for abababa, when the tape is ␣␣␣b␣␣␣ then replace b with ␣ & goto final state.



Ref: https://youtu.be/4kLxD12gbyA?si=y35w8H7bU38tJjms

# Turing machine for $L=\{wcw \mid w \in \{a,b\}^*\}$

Algorithm:

I) Replace the first a with X or first b with Y & move to Right.

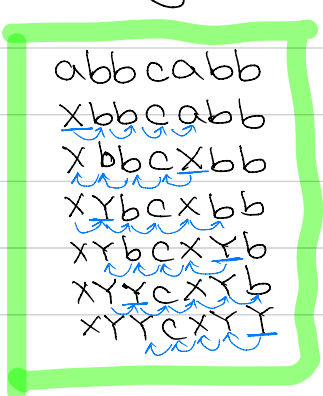
II) Continue moving right until we reach first a or b which is situated after c.

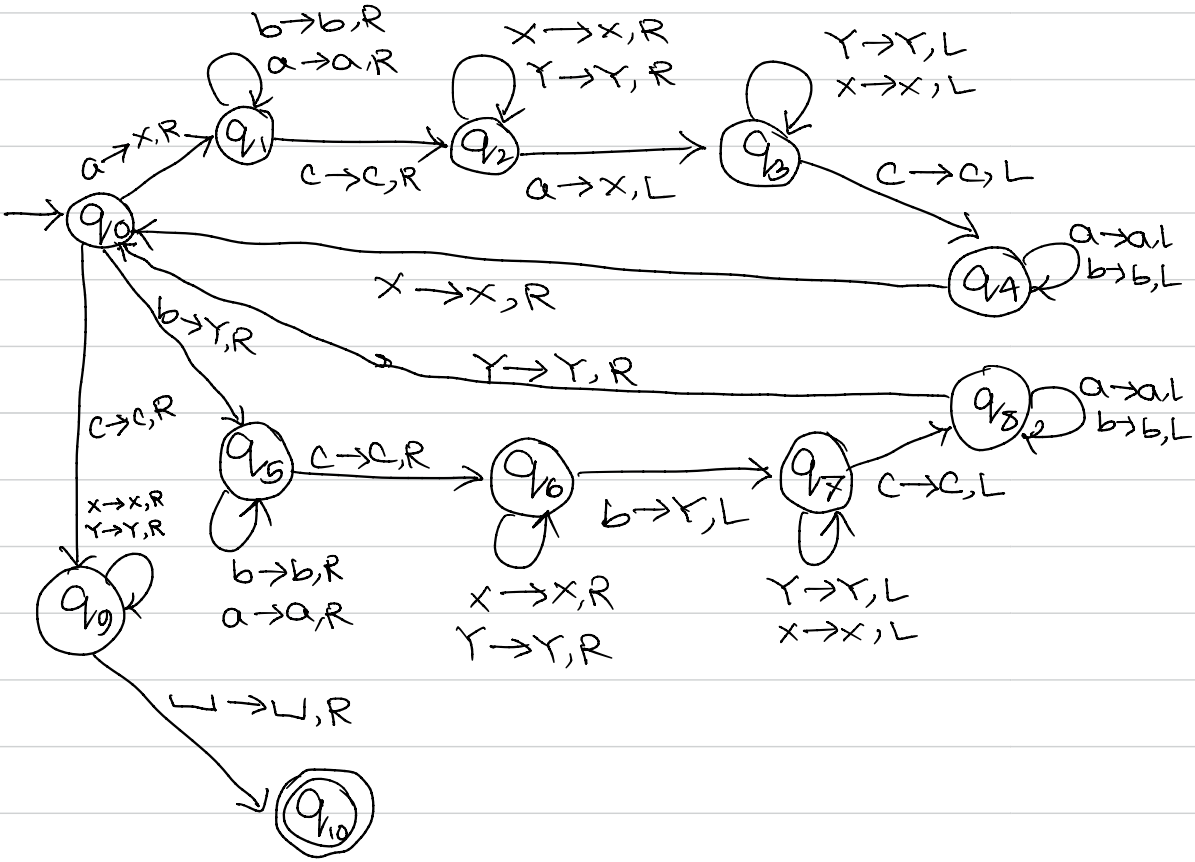
III) Replace the first a/b after c with X/Y & move left.

IV) Keep moving left until we reach the first a or b which is situated before c.

V) Continue the same process until there is no a/b left.

VI) Keep c as it is & reach the end of the string to reach final state.




q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
q10
a→X,R
b→b,R
a→a,R
c→c,R
X→X,R
Y→Y,R
a→X,L
Y→Y,L
X→X,L
c→c,L
a→a,L
b→b,L
X→X,R
b→Y,R
Y→Y,R
c→c,R
c→c,R
X→X,R
Y→Y,R
b→b,R
a→a,R
b→Y,L
X→X,R
Y→Y,R
c→c,L
Y→Y,L
X→X,L
a→a,L
b→b,L
␣→␣,R