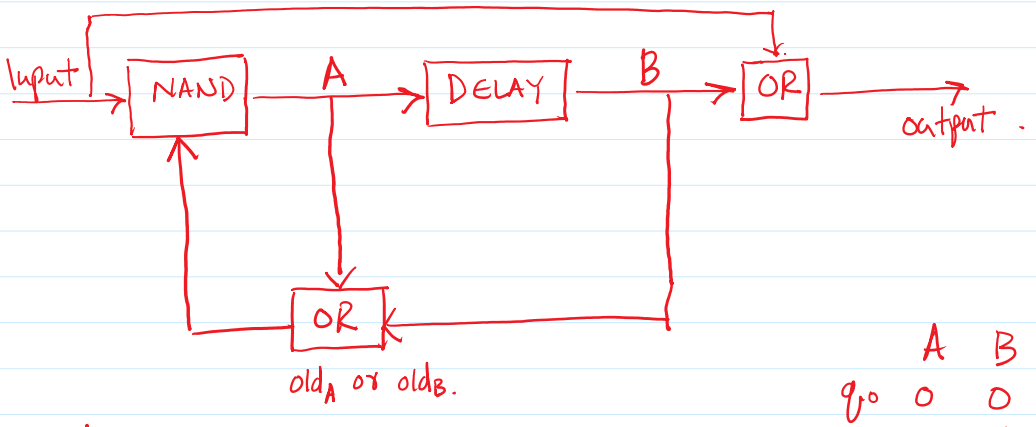
# Lecture 17 :- Finite Automata In Sequential Circuits. (Mealy Machine).

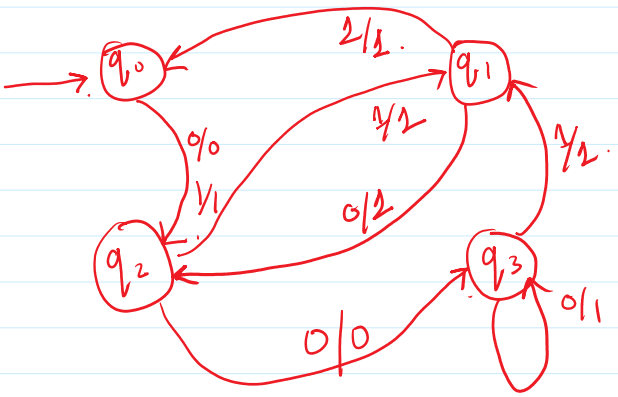
| A | B | A.B | ¬(A.B) |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |



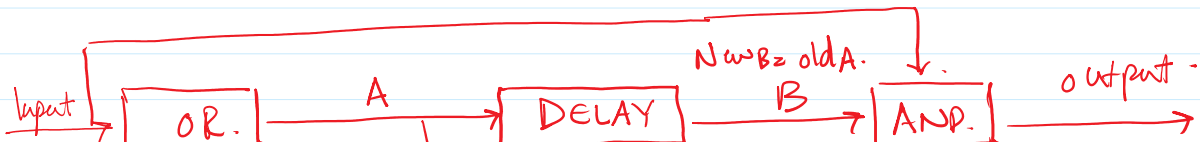

NewB = OldA.
NewA = Input NAND. (oldA or oldB)
Output = OldB or Input.

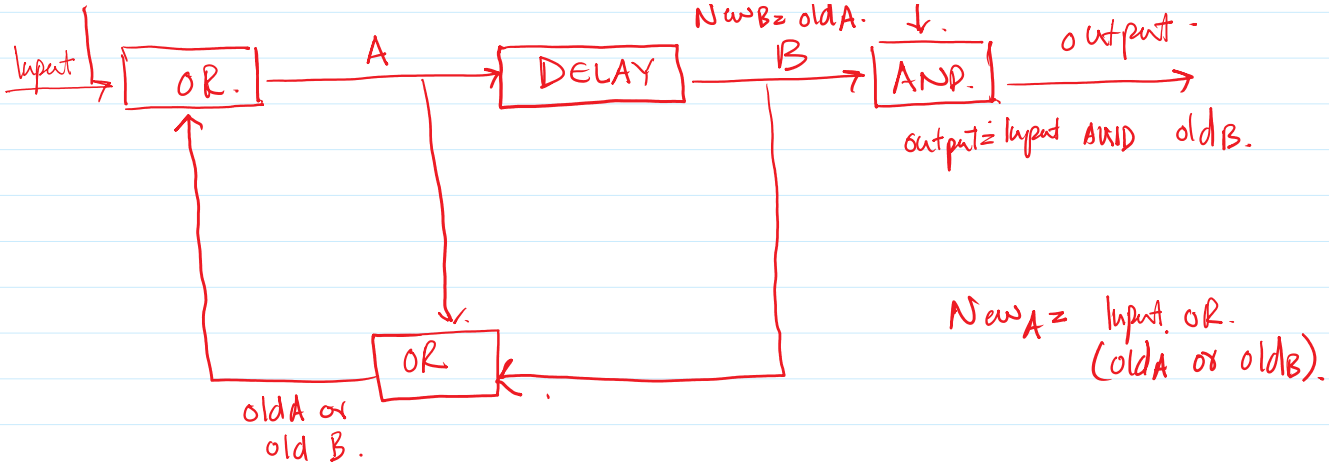
| | A | B |
|---|---|---|
| $q_0$ | 0 | 0 |
| $q_1$ | 0 | 1 |
| $q_2$ | 1 | 0 |
| $q_3$ | 1 | 1 |

| | old State | | Input 0 New State | | output | Input 1. New State | | output |
|---|---|---|---|---|---|---|---|---|
| | oldA | oldB | NewA | NewB | | NewA | NewB | |
| → $q_0$ | 0 | 0 | (1 | 0) $q_2$ | 0 | (1 | 0) $q_2$ | 1 |
| $q_1$ | 0 | 1 | (1 | 0) $q_2$ | 1 | (0 | 0) $q_0$ | 1 |
| → $q_2$ | 1 | 0 | (1 | 1) $q_3$ | 0 | (0 | 1) $q_1$ | 1 |
| $q_3$ | 1 | 1 | (1 | 1) $q_3$ | 1 | (0 | 1) $q_1$ | 1 |




| A | B | A and B | A NAND B |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |





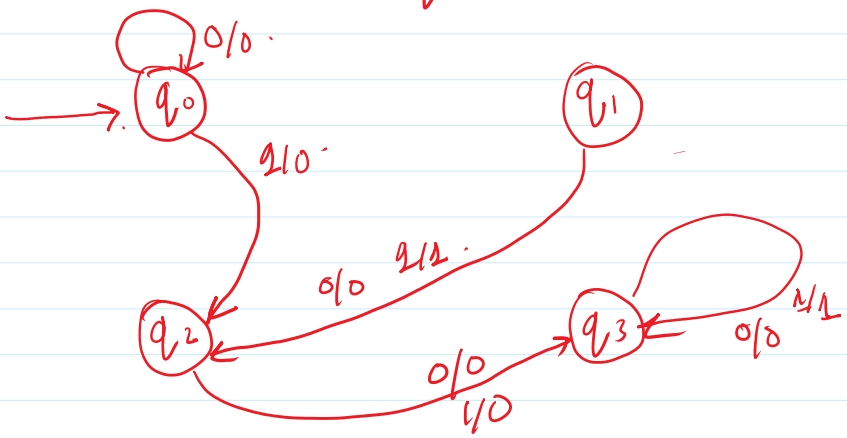

New$_A$ = Input or (old$_A$ or old$_B$).

New$_B$ = old A

output = Input AND old$_B$.

| | A | B. |
|---|---|---|
| $q_0$ | 0 | 0 |
| $q_1$ | 0 | 1 |
| $q_2$ | 1 | 0 |
| $q_3$. | 1 | 1. |

| | Old State | | INPUT 0 | | | INPUT 1. | | |
|---|---|---|---|---|---|---|---|---|
| | | | New State | | Output | New State | | Output |
| | old$_A$ | old$_B$ | New$_A$ | New$_B$ | | New$_A$ | New$_B$. | |
| $q_0$ | 0 | 0 | (0 | 0) $q_0$ | 0 | (1 | 0) $q_2$ | 0 |
| $q_1$ | 0 | 1 | (1 | 0) $q_2$ | 0 | (1 | 0) $q_2$. | 1. |
| $q_2$ | 1 | 0 | (1. | 1) $q_3$ | 0 | (1. | 1) $q_3$ | 0 |
| $q_3$. | 1 | 1. | (1 | 1) $q_3$ | 1 | (1. | 1) $q_3$. | 1. |




REGULAR LANGUAGES :-

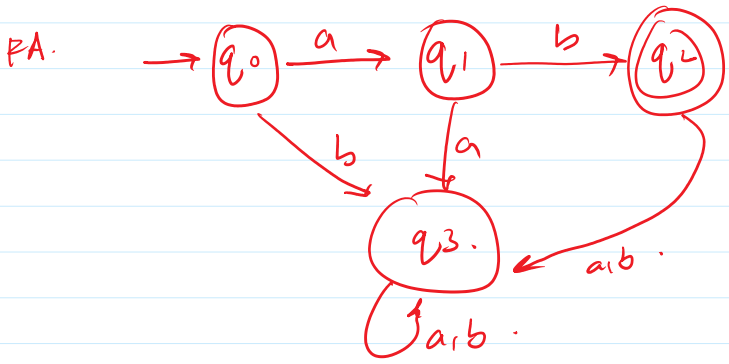
A language is called Regular if ∃ a Regular Expression defining the Language

palindrome.

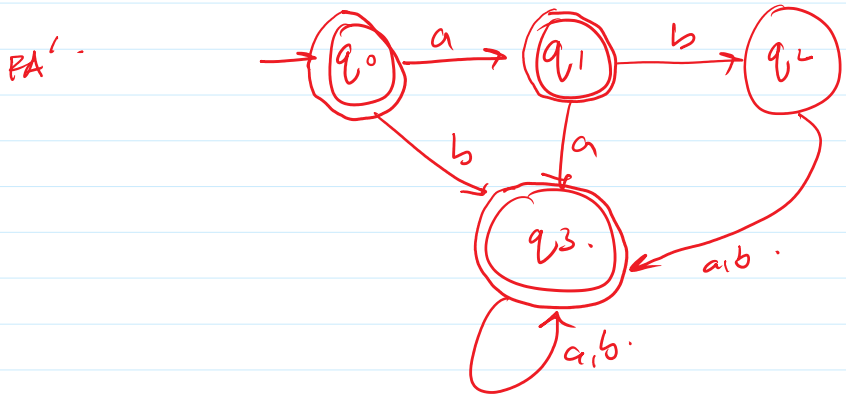
Properties :-
1- closure.
2- Complement & Intersection.

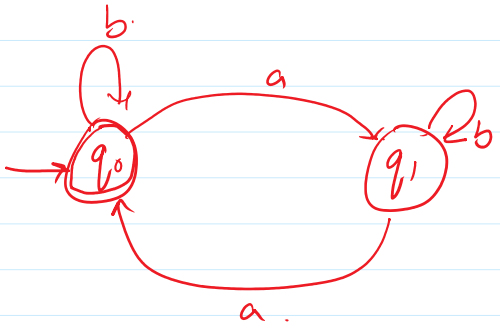
PA, let PA' be the Complement.
PA' = All the words that are not accepted by PA.

Procedure for determining a Complement.
HW. { 1- change all final states to Non final.
2- change all non final to final.

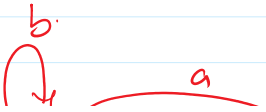



ab.

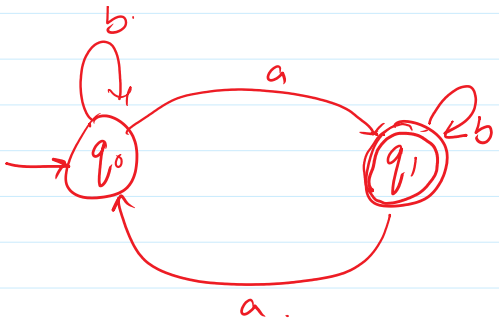



a's even.




PA'. odd a's




DFA' .

odd a's