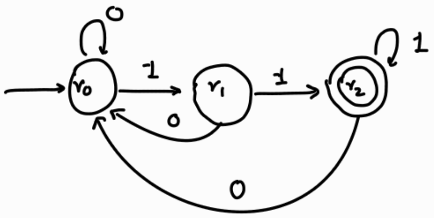
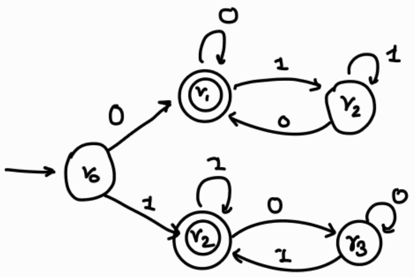


DFA

- r_1 is remembering information, it cannot be reached without a '1'
- r_2 similarly, 2 consequtive 1's.



← Starting and ending with same symbol.

1. Can a computer Solve every problem?

⇒

- What is a Problem?
  - → Every language is a problem you're trying to solve.
  - Every subset of Σ^* is a problem.

Alphabet ⇒ $\Sigma = \{0,1\}$

$\Sigma^* = \{\epsilon, 0, 1, 00, 01 \ldots\}$

Language ⇒ $L \subseteq \Sigma^*$

Regular Language ⇒ A language is regular if some DFA accepts/recognises it }

- DFA ⇒ 5-tuple → $Q, \Sigma, \delta, q_0, F$

Q → finite set of all states

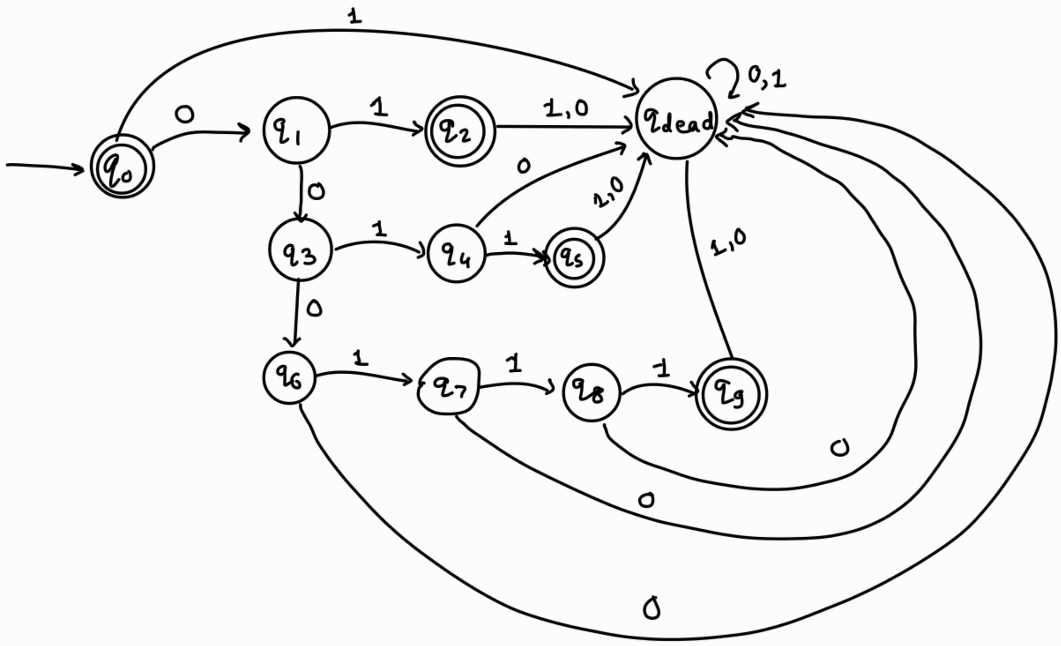
Σ → Alphabet

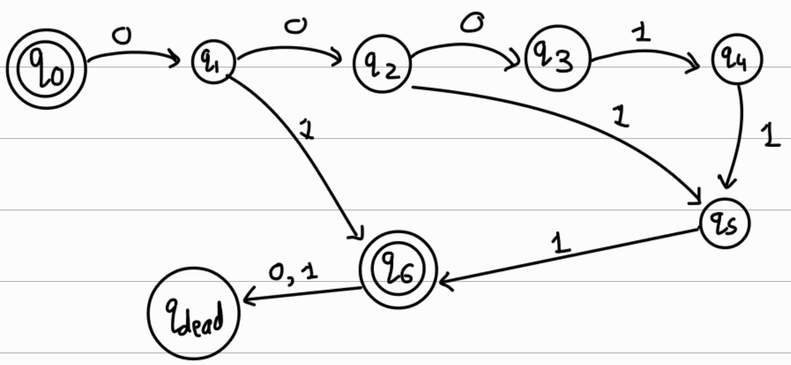
δ → Transition Function → $\delta \rightarrow Q \times \Sigma$

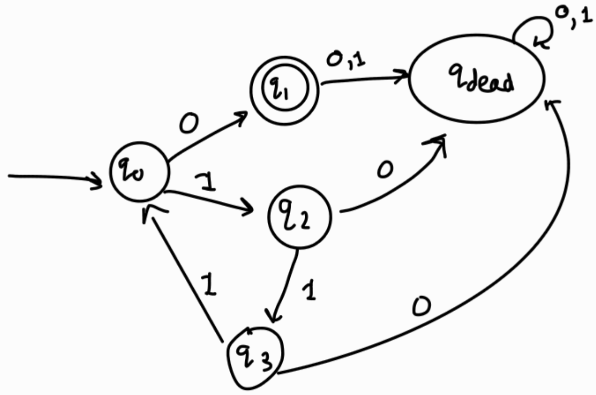
q_0 → Element of Q, $q_0 \in F$, initial state

F → Set of Final states, $F \subseteq Q$

$L_3 = \{0^n 1^n \mid n \leq 3\}$







$L = \{ w \mid w = 0 1^{3n} 0 \ , \ n \geq 0 \}$

p will always be less than no. of states, due to loops & pigeonhole principle.

string = 01110
$x = 0$
$y = 111$
$z = 0$

OR

$s = 0111110$
$x = 0$
$y = 111$
$z = 1110$

$p = 5$, as we can find y in the string of size 5.

* $x \rightarrow$ takes you to loop
$y \rightarrow$ loop part
$z \rightarrow$ after loop wala part till final state.

$s \in A \rightarrow$ every string in the language A

If A is a regular language then there exists a number p (pumping length), show that every $s \in A$ of length at least p can be divided into 3 pieces/parts, x, y and z, satisfying following conditions:

$xy^iz \in A \qquad i \geq 0$

$|y| > 0$

$|xy| \leq p$

∴ we call this pumping as we are pumping 'y' i times, initially it was only 1.

$(p = |Q|?)$

Pigeonhole principle

S.T. language is not regular!

Q) $L5 = \{0^n 1^n \mid n \geq 0\}$

→ Proof by contradiction:

Assume 'L5' is regular language.

- As L5 is regular, there is a DFA,

$M5 = \{Q, \Sigma, \delta, q_0, F\}$

- By pumping lemma, there exist some pumping length P,

$P = |Q|$ ... by Pigeonhole principle.

∴ A string from language will be $0^P 1^P$,

we know, for Pumping lemma, $|y| > 0$ and $|xy| \leq P$,

∴ if we assume $x = \epsilon$,

$\underline{y = 0^P}$ ∴ $y > 0$ and $|xy| \underline{\leq} P$,

but if we pump 'y' in xyz,

$xy^2z \notin L5$, since no. of zeroes > no. of ones.

∴ we arrive at contradiction, as for regular languages, y can be pumped i times, with $i > 0$, and $xy^iz \in L5$, but that's not the case.

Q) $L6 = \{ww \mid w \in \Sigma^*\}$ Prove Non-regular.

- Assume L6 is regular.
- ∴ There exist DFA,

$M_6 = \{Q, \Sigma, \delta, q_0, F\}$

- By pumping lemma,

$p = |Q|$ ... pigeonhole principle.

$S = 0^P 1 0^P$ ... we can't have $0^P 0^P$ as they can be pumped!

$x = \epsilon$, $y = 0^P$, $|xy| \leq P$

$|y| > 0$

but $xy^iz \notin L6$

$L_7 = \{ 1^{n^2} \mid n \geq 0 \}$

$\therefore L_7 = \{ \epsilon, 1, 1111, 111111111 .. \}$

$M_7 = \{ Q, \Sigma, \delta, q_0, F \}$

By pumping lemma, there exists $P = |Q|$,

$S = 1^{P^2}$

the next string will be

$S_2 = 1^{(P+1)^2}$

$= 1^{P^2 + 2P + 1}$

Now, we have a condition that '$y \leq P$', therefore however we pump 'y', it will always fall short of the next string in language.

$\therefore xy^iz \notin L_7$

$\therefore$ we arrive at contradiction,

$\therefore L_7$ is not regular.

PS //
Do check the methodology that the faculty tells in class
. xD