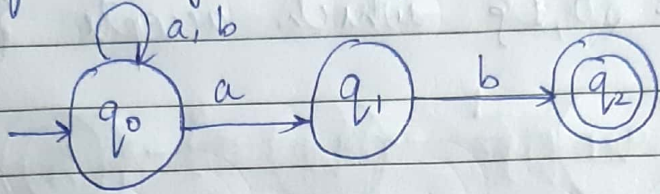
31. decimal ~~[illegible]~~ divisible by 3.

$r = 10$ $i = 0, 1, 2$ $d = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ $K = 3$.

remainder digits.

NFA – More number of transition function (or states) for a given input alphabet.

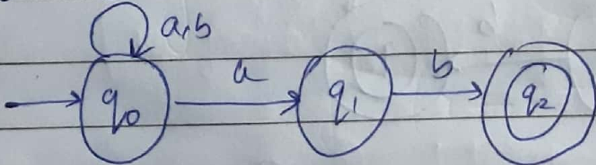
for eg:-



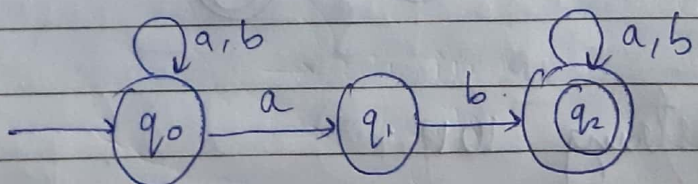
Note:- We don't have to find transition state for every input alphabet over a particular state.

Examples.

1. Construct a ~~D~~NFA with input alphabet $\Sigma = \{a, b\}$ that ends with a, b $L = \{aab, ab, bab, bbab, aaab \ldots\}$.

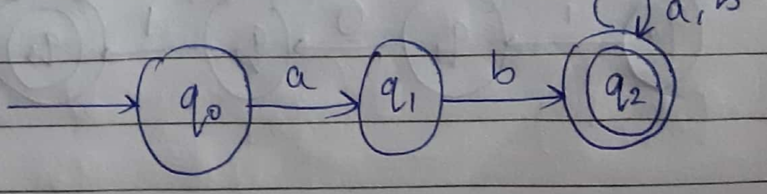


2. CNFA ~~ends~~ has substring 'ab'.



δ	a	b
q_0	$\{q_0, q_1\}$	q_0
q_1	ϕ	q_2
q_2	q_2	q_2

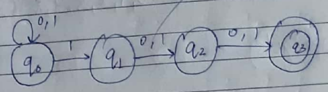
3. CNFA starts with 'ab'



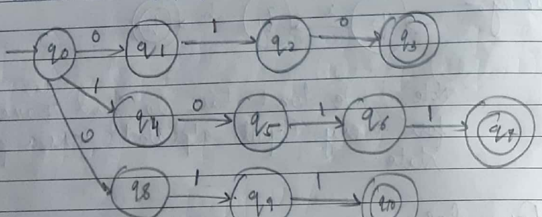
δ	a	b
q_0	q_1	ϕ
q_1	ϕ	q_2
$*q_2$	q_2	q_2

Scanned with CamScanner

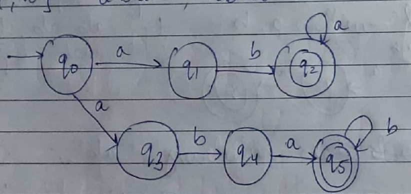
4. CNFA over {0,1} such that 3rd symbol from right end should be 1.



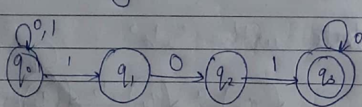
5. CNFA over {0,1} which accepts {0101, 1011, 011}



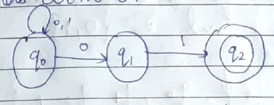
6. {a,b} aba^n, $abab^n$



7. Having substring 101.



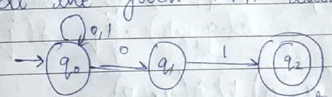
8. Ends with 01



Conversion of NFA to DFA.

1. Subset construction method
2. Lazy evaluation method.

* Convert the given NFA into its equivalent DFA using subset construction method.



1st Step: In the given NFA, the start state and input alphabet for NFA remains same in DFA.

2nd step: NFA → n states
DFA → 2^n states

3rd step: Power set = {ϕ, q_0, q_1, q_2, {q_0, q_1}, {q_1, q_2}, {q_0, q_2}, {q_0, q_1, q_2}}.

4th step → The final state of NFA if contained in the subset of the power set, then that particular subset becomes the final state in DFA.

5th step → (If the set contains more states)

$$\delta(\{q_0,q_1\},0) = \delta(q_0,0) \cup \delta(q_1,0)$$
$$= \{q_0,q_1\} \cup \phi$$
$$= \{q_0,q_1\}$$

δ	0	1
q_0	$\{q_0, q_1\}$	q_0
q_1	ϕ	q_2
$^*q_2$	ϕ	ϕ

1. q_0 is start state in DFA and $\Sigma = \{0,1\}$

2. $\{\phi, q_0, q_1, q_2, \{q_0, q_1\}, \{q_1, q_2\}\ \{q_0, q_2\}\ \{q_0, q_1, q_2\}\}$

3. Transition states :-

$\delta(q_0, 0) = \{q_0, q_1\}$ $\quad \delta(q_1, 0) = \phi$ $\quad \delta(q_2, 0) = \phi$

$\delta(q_0, 1) = q_0$ $\quad \delta(q_1, 1) = q_2$ $\quad \delta(q_2, 1) = \phi$

$$\delta(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \phi = \{q_0, q_1\}$$

$$\delta(\{q_0, q_1\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) = q_0 \cup q_2 = \{q_0, q_2\}$$

$$\delta(\{q_0, q_2\}, 0) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \phi = \{q_0, q_1\}$$

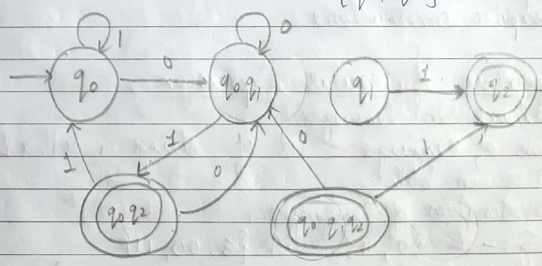
$$\delta(\{q_0, q_2\}, 1) = \delta(q_0, 1) \cup \delta(q_2, 1) = q_0 \cup \phi = q_0$$

$$\delta(\{q_1, q_2\}, 0) = \delta(q_1, 0) \cup \delta(q_2, 0) = \phi \cup \phi = \phi$$

$$\delta(\{q_1, q_2\}, 1) = \delta(q_1, 1) \cup \delta(q_2, 1) = q_2 \cup \phi = q_2$$

$$\delta(\{q_0, q_1, q_2\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0)\ \delta(q_2, 0) = \{q_0, q_1\} \cup \phi \cup \phi = \{q_0, q_1\}$$

$$\delta(\{q_0, q_1, q_2\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1) = q_0 \cup \phi \cup q_2 = \{q_0, q_2\}$$



→ Lazy Evaluation Method

→ Transition states :-

$\delta(q_0, 0) = \{q_0, q_1\}$

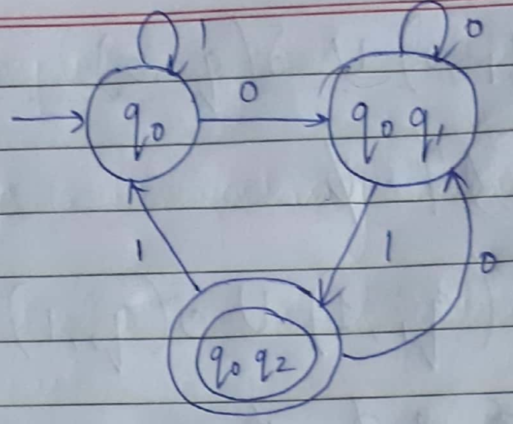
$\delta(q_0, 1) = q_0$

$\delta(\{q_0, q_1\}, 0) = \{q_0, q_1\}$

$\delta(\{q_0, q_1\}, 1) = \{q_0, q_2\}$

$\delta(\{q_0, q_2\}, 0) = \{q_0, q_1\}$

$\delta(\{q_0, q_2\}, 1) = q_0$



UNIT – II

* Regular Expressions → Algebraic-like description to denote Regular Language.

* Applications of Regular Expressions.
- UNIX.
- 'grep' command.
- Compiler constructions.

* Operators of regular Expression.

→ Operations on Languages.

→ Union of Languages – Set of strings in either L, or in M or in both.

→ Concatenation of languages.

Eg :– $L = \{0, 11\}$ $M = \{\varepsilon, 00, 11\}$

$L \cdot M = \{0, 11, 000, 011, 1100, 1111\}$

→ The closure (star or Kleene closure) of a language

$L = \{0, 1\}$ $L^* = \{$set of all strings of any length$\}$

$L^0 = \{\varepsilon\}$

$L^1 = \{0, 1\}$

$L^2 = \{00, 01, 10, 11\}$

$L = \{0, 11\}$ $L^2 = \{011, 110, 1111, 00\}$

$L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \ldots$