In the previous lecture, we were trying to show the following.

Theorem 1.26: Regular languages are closed under the concatenation operation.

We needed a new ingredient.

Nondeterministic Finite Automata (NFA)

Let us see the major differences between DFA and NFA.

1. DFA has exactly one outgoing arrow / transition for each state $s \in Q$ and symbol $a \in \Sigma$.

NFA can have 0, 1 or more than 1.

2. DFA has all the arrows labelled with symbols in the alphabet.

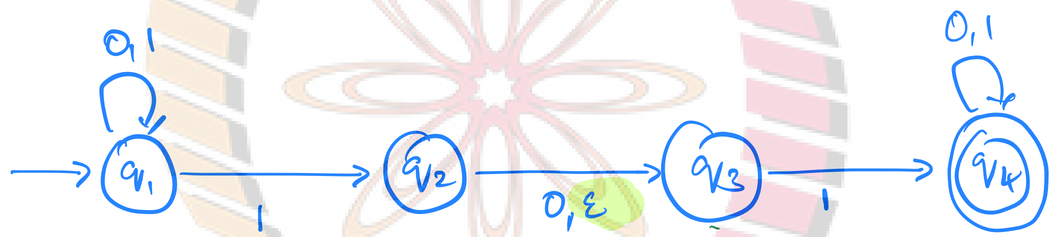
empty string

NFA can have arrows marked with ε.

Can have 0, 1 or many such arrows.

3. NFA can have multiple choices to be made at each state, possibly. It could have many possible computation paths.

4. NFA accepts a string if there is at least one accepting computation path.



1111 — Accepted by the NFA above.

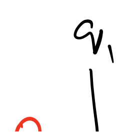
100 — Is not accepted.

1010 — Accepted

1 $q_1 \to q_2$
ε $q_2 \to q_3$
1 $q_3 \to q_4$
1 $q_4 \to q_4$
1 $q_4 \to q_4$

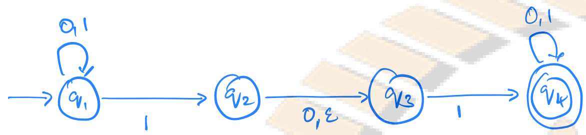
$L = \{ w \in \{0,1\}^* \mid$ w contains 101 or 11 as a substring $\}$

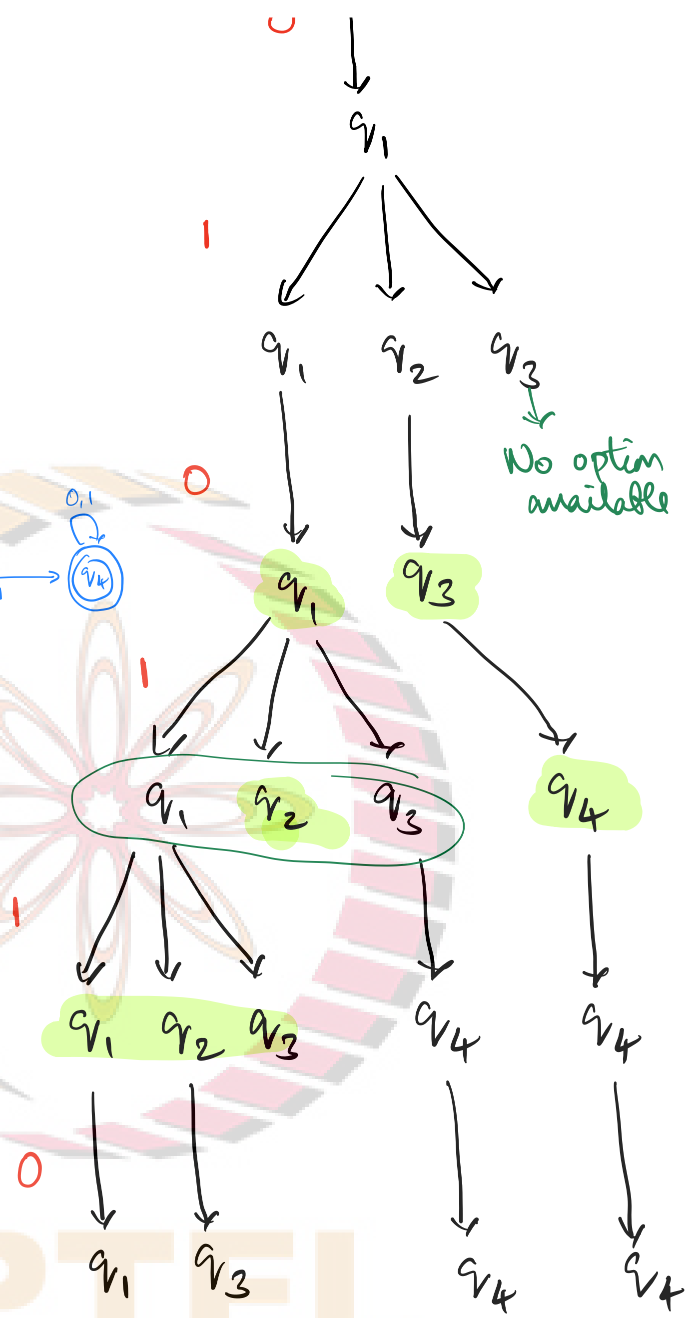
5. Nondeterminism can be viewed as guessing.



0 1 0 1 1 1 0

0 1 0 1 1 1 0



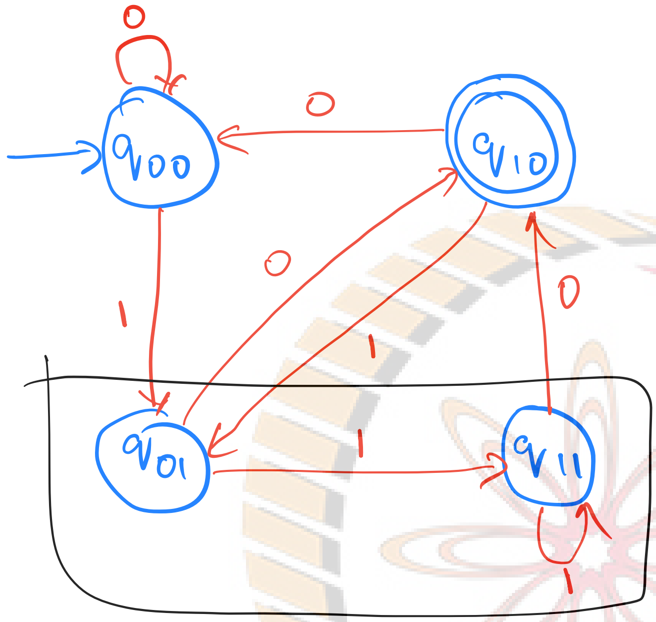


NFA's are at least as powerful as DFA's.

But we may be able to do the computation using a smaller no. of states.

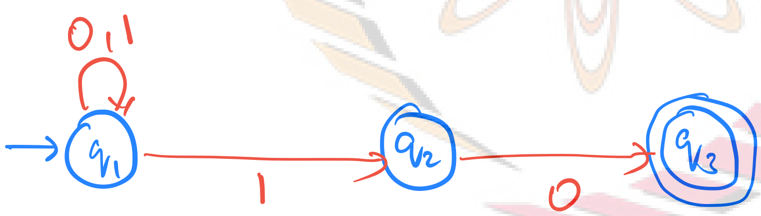
Examples

$A = \{ w \mid w \text{ ends in } 10 \}$



DFA for A

0011010

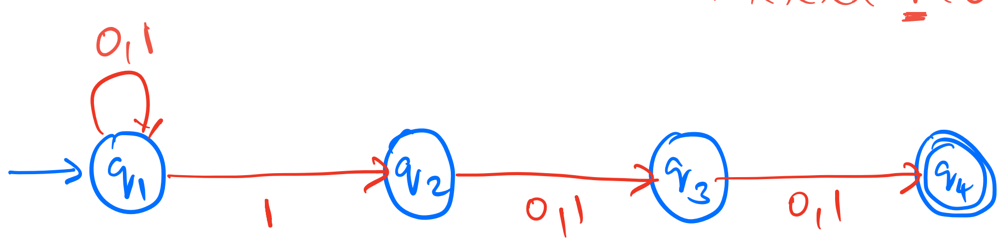


NFA

NPTEL

Example 1.30: Language of strings where the third last symbol is a 1.

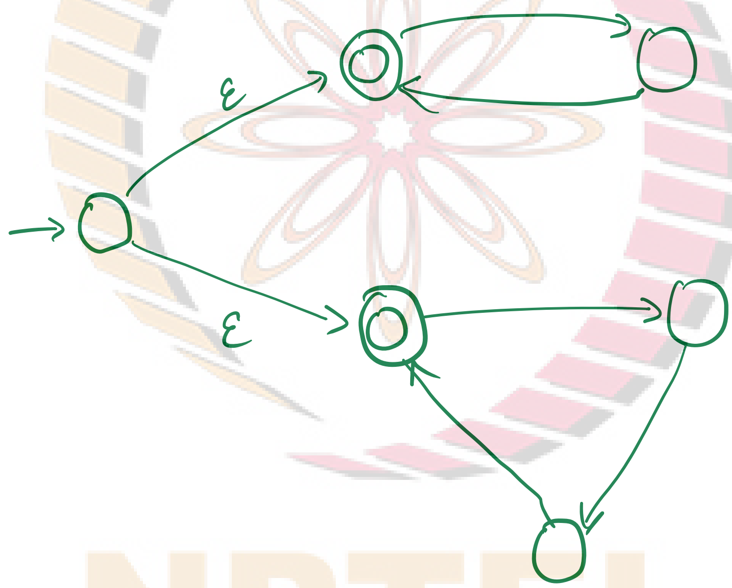
x x x x x 110



Read the book to see the DFA. (has 8 states!)

Example 1.33: Unary language.

$A = \{ 0^k \mid k \text{ is a multiple of } 2 \text{ or } 3 \}$

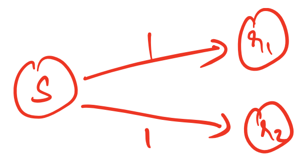


Def 1.37: A nondeterministic finite automaton (NFA) is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$

1) Q is a finite set of states

$\delta(s,1) = \{r_1, r_2\}$

2) Σ is a finite alphabet



3) $\delta : Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function

[$\delta : Q \times \Sigma \to Q$] in DFA

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

$\hookrightarrow$ Power set of Q.

$\quad \parallel$

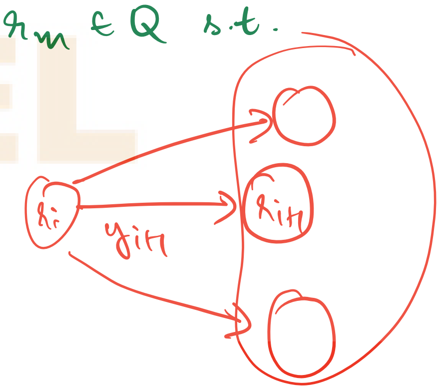
Set of all subsets of Q.

$\delta(s, \varepsilon) = \{q_1\}$

4) $q_0 \in Q$ is the start state

5) $F \subseteq Q$ is the set of accepting states.

NFA computation : NFA N accepts w if we can write $w = y_1 y_2 \dots y_m$ where $y_i \in \Sigma_\varepsilon$ and a sequence of states $r_0, r_1, \dots r_m \in Q$ s.t.



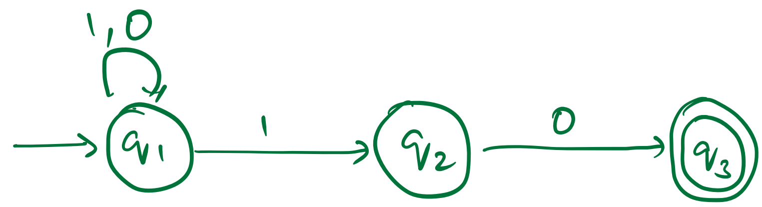
(1) $r_0 = q_0$

(2) $r_{i+1} \in \delta(r_i, y_{i+1})$

for $0 \le i \le m-1$

(3) $r_m \in F$.

Example:



$Q = \{q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

$F = \{q_3\}$

Start State $= q_1$

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

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

$\delta(q_1, \varepsilon) = \phi$

$\delta(q_2, 0) = \{q_3\}$

$\delta(q_2, 1) = \phi$

$\delta(q_3, 0) = \delta(q_3, 1) = \phi.$

Read Example 1.38 in the book.