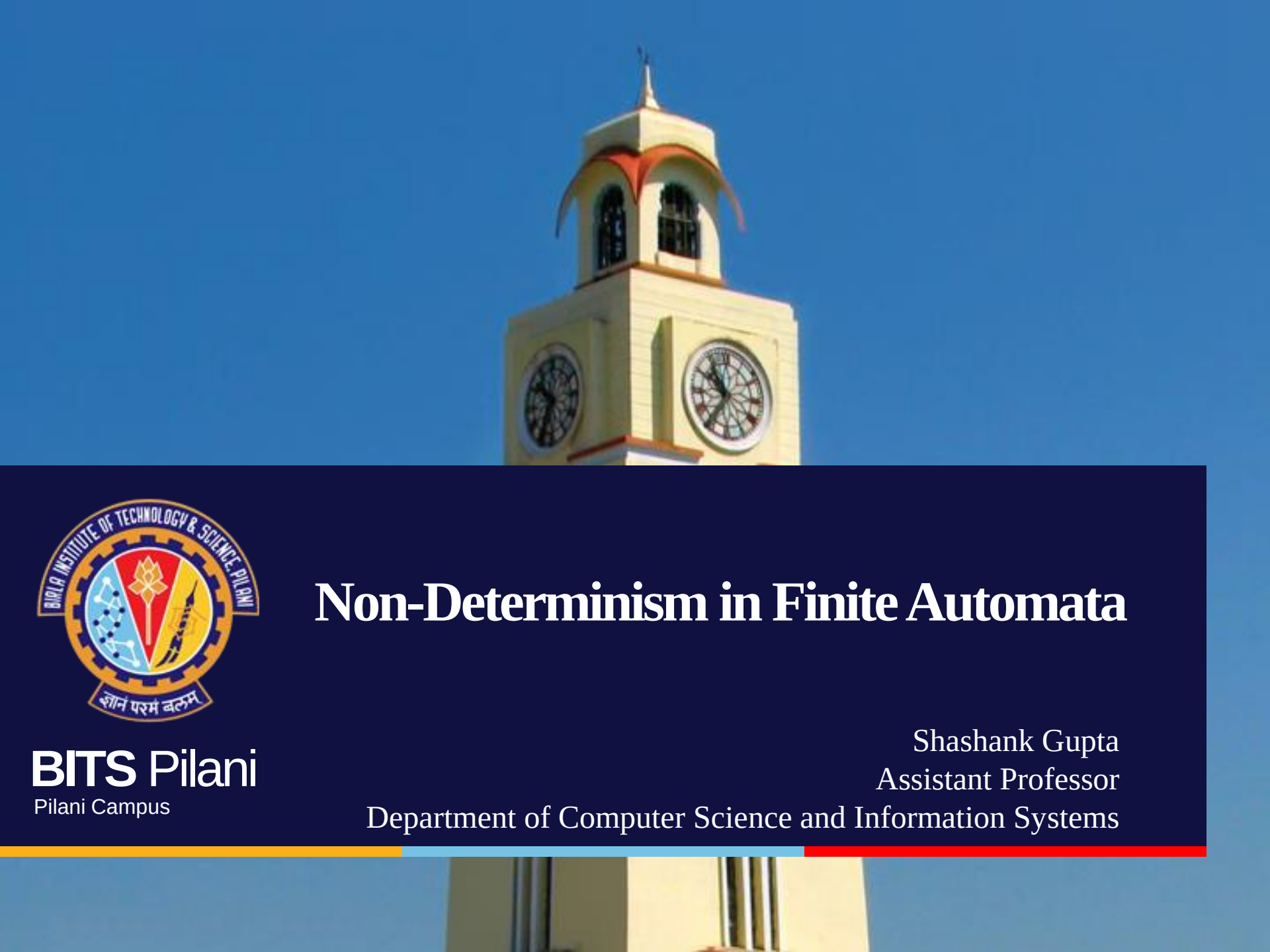


Non-Determinism in Finite Automata

BITS Pilani
Pilani Campus

Shashank Gupta
Assistant Professor
Department of Computer Science and Information Systems

Non-Deterministic Finite Automata (NFA)

Non-Determinism

From a state q on an input symbol a, the automaton can go to multiple k states **(k >= 0)**

Here, computation executes simultaneously along each of these paths.

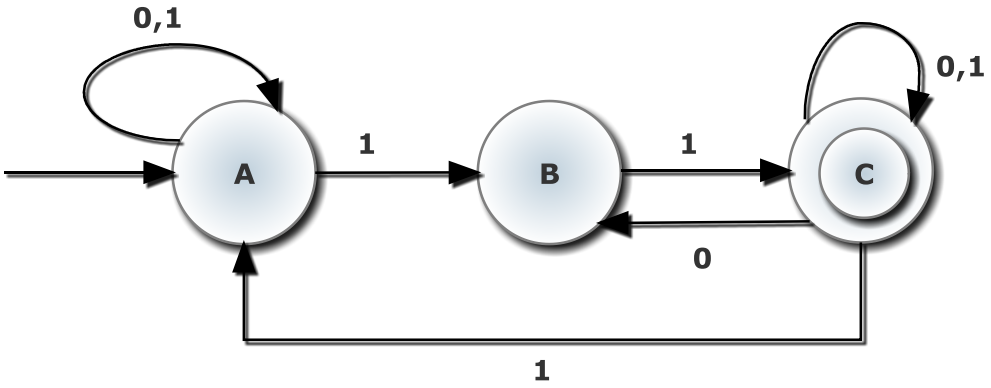
Non-Deterministic Finite Automata (NFA)

In NFA, from each state, there can be any number of transitions over Σ.

There is no chance of having dump/dead state in NFA, unlike in DFA.

Example of NFA

- Consider a NFA over $\Sigma = \{0,1\}$



Formal Definition of NFA

- It is represented by 5 tuples $N = (Q, \Sigma, \delta, q_0, F)$ where
  - Q: Finite Set of States
  - Σ: Input Alphabet
  - δ: Transition Function i.e. $Q \times \Sigma \rightarrow 2^Q$
  - $q_0 \in Q$ is the Start State
  - $F \subseteq Q$ is the Set of Accept States.

  A NFA $N = (Q, \Sigma, \delta, q_0, F)$ accepts a string w by one of the computation path leads to one of the accept state.

  Hence, NFA will reject the strings only if all the computation paths ends up in any of the non-accept states.

Non-Deterministic Finite Automata (NFA)

Now, according to the definition of the NFA i.e. N = {Q, ∑, δ, q0, F}, here

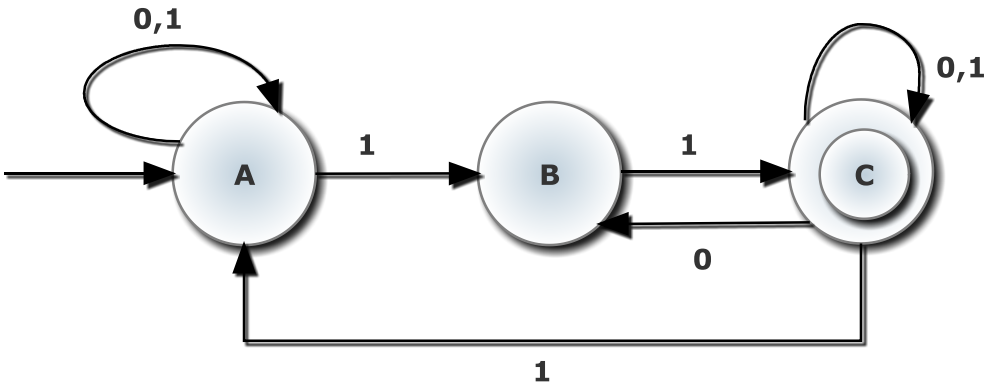
-Q: {A, B, C}

-∑: {0, 1}

- δ : δ(A,0) → {A}
  δ(A,1) → {A, B}
  δ(B,0) → {} or ϕ
  δ(B,1) → {C}
  δ(C,0) → {B, C}
  δ(C,1) → {A, C}

-q0: A

-F: {C}



Designing Examples of NFA

Construct a NFA which accepts all strings of 0's and 1's where the second input symbol from RHS is '0'.

NFA = ?

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

Example (Continued…..)

Construct a NFA which accepts all strings of 0's and 1's where the second input symbol from RHS is '0'.

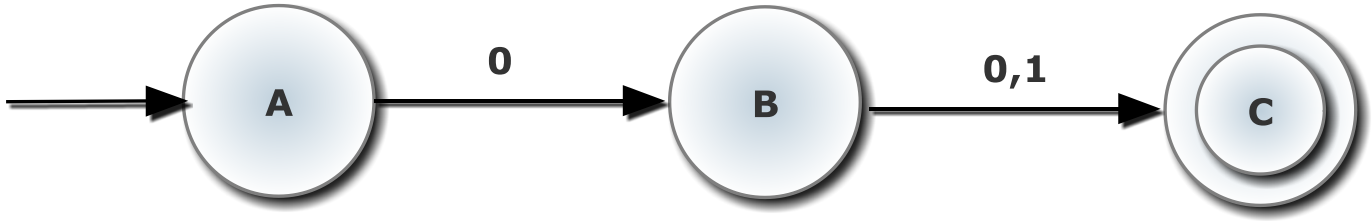
Step 1: Initially, construct the language 'L' over Σ = {0, 1} starting with the string of minimum length.

- L = {**0**1, **0**0, 101**0**1, 0111**0**1, 000**0**0, -------}

Example (Continued…..)

Step 2: Secondly construct the FA for the minimal length string from the language ‘L’.

- L = {01, 00, 10101, 01101, 00000, ------}



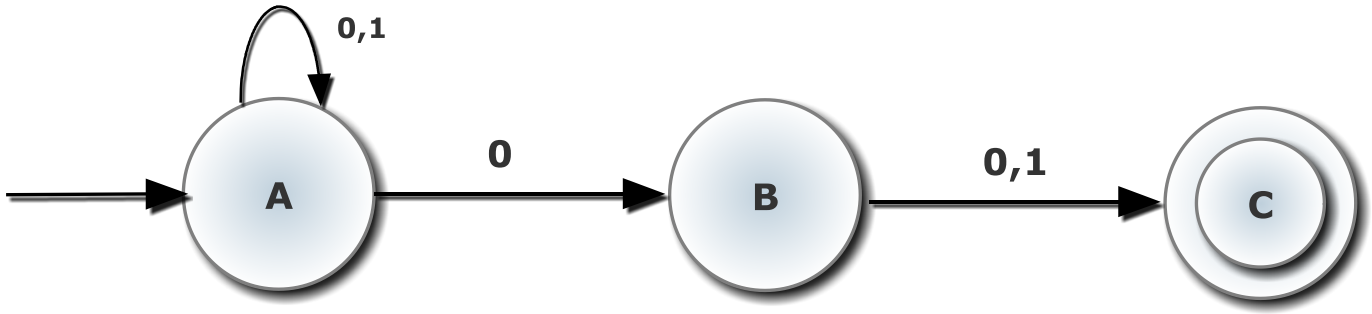
Home Assignment

- Construct the NFA's for all the possible languages for which we recently design DFA's.

Example (Continued…..)

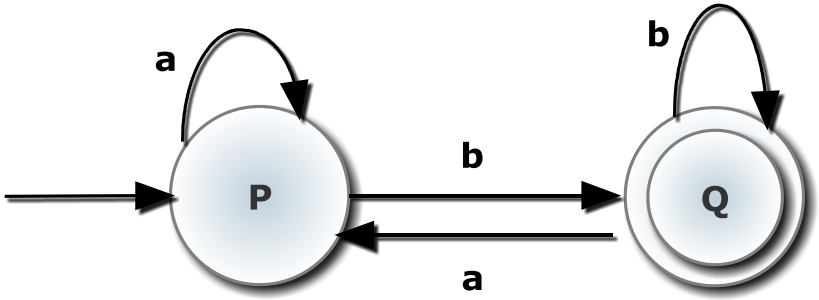
Step 3: Finally, transform the FA into NFA by considering all the strings specified in the Language ‘L’.

- L = {**0**1, **0**0, 101**0**1, 0111**0**1, 0000, ------}



Equivalence of NFA and DFA

Transition Table Notation of FA

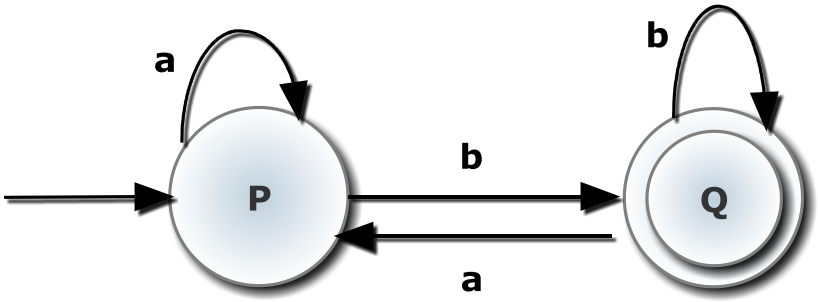


STATES	NOTATIONS
Initial State	
Intermediate State	q
Final State	

Transition Table Notation (Continued…..)

innovate achieve lead

Graphical Notation of FA



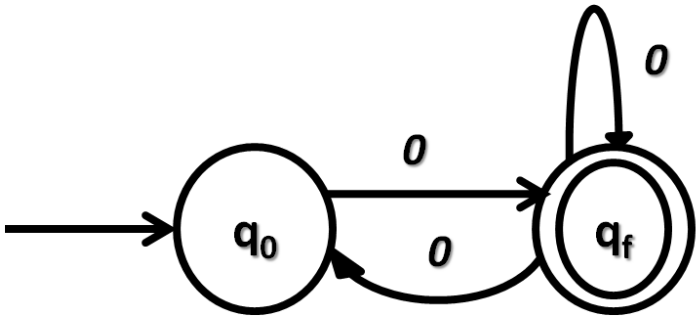
Transition Table Notation of FA

I/P Symbol / STATE	a	b
→ P	P	Q
Q	P	Q

Equivalence of NFA and DFA

Construct an equivalent DFA for the given NFA over the input alphabet $\sum = \{0, 1\}$

Graphical Notation of NFA



Transition Table Notation of NFA

I/P Symbol / STATE	0	1
$\rightarrow q_0$	$\{q_f\}$	Φ
q_f	$\{q_0, q_f\}$	Φ

Equivalence of NFA and DFA

Transition Table Notation for NFA

I/P Symbol / STATE	0	1
$\rightarrow q_0$	$\{q_f\}$	Φ
q_f	$\{q_0, q_f\}$	Φ

Transition Table Notation for DFA

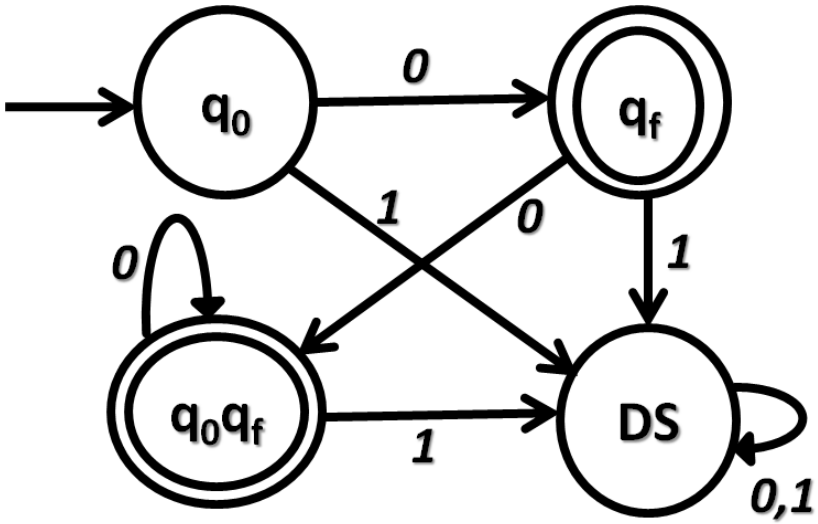
I/P Symbol / STATE	0	1
$\rightarrow q_0$	q_f	DS
q_f	q_0q_f	DS
DS	DS	DS
q_0q_f	$[q_0q_f]$	DS

Equivalence of NFA and DFA

Transition Table Notation for DFA

I/P Symbol / STATE	0	1
→q_0	q_f	DS
(q_f)	q_0q_f	DS
DS	DS	DS
(q_0q_f)	$[q_0q_f]$	DS

Equivalent Graphical Notation for DFA



Equivalence of NFA and DFA

For every NFA, there is an equivalent DFA.

Theorem: Let $N = (Q, \Sigma, \delta, q_0, F)$ be a NFA.

- There exists a DFA $D = (Q`, \Sigma, \delta`, q_0`, F`)$ s.t. $L(N) = L(D)$

Construction of DFA from NFA

$$Q` = 2^Q$$

$$Let\ A \subseteq Q\ then\ \delta(A,a) = \bigcup_{r \in A} \delta(r,a)$$

$$q`_0 = \{q_0\}$$

- $F` = \{A \subseteq Q \mid A \cap F \neq \phi\}$