Session 10

Nondeterministic Finite Automata with Λ-transitions

NFA-Λ

- Motivation & concept
- Definition
- Λ-closure of a set of states
- Extended Transition Function for NDF-Λ
- Converting NFA-Λ into NFA
- Converting FA into NFA-Λ

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

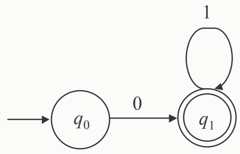
Motivation

- If there are machines M_1 and M_2 accepting L_1 and L_2 there are machines M_u, M_i *and* M_d accepting:
  - L(M_u) = $L_1 \cup L_2$
  - L(M_i) = $L_1 \cap L_2$
  - L(M_d) = $L_1 - L_2$
- What about machines M_c and M_k for the inductive part of the definition of *RE*?
  - L(M_u) = $L_1 \cup L_2$
  - L(M_c) = L_1L_2
  - L(M_k) = L_1^*

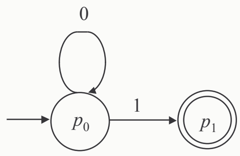
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

First: union and concatenation

- $\Sigma = \{0, 1\}$
- L_1 = $\mathbf{01^*}$ = $\{0\}\{1\}^*$



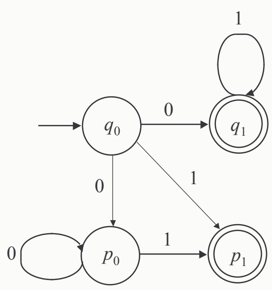
- L_2 = $\mathbf{0^*1}$ = $\{0\}^*\{1\}$



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

The union, again

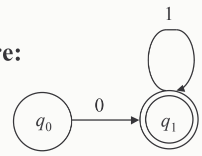
- $L_1 \cup L_2$ = $\mathbf{01^* + 0^*1}$ = $\{0\}\{1\}^* \cup \{0\}^*\{1\}$

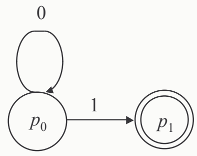


Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Union

- $L_1 \cup L_2$ = $\mathbf{01^* + 0^*1}$
- **The original structure:**

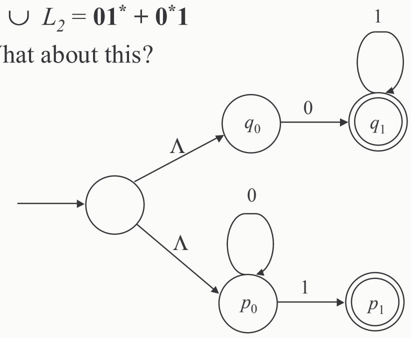




Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

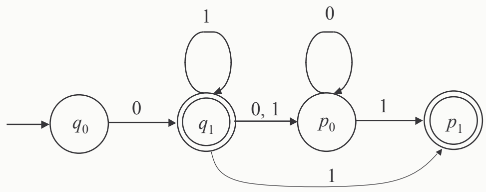
Union

- $L_1 \cup L_2 = \mathbf{01^* + 0^*1}$
- What about this?



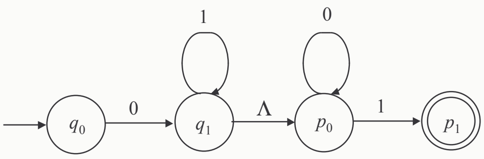
Concatenation

- $L_1 \cup L_2 = \mathbf{01^*0^*1}$



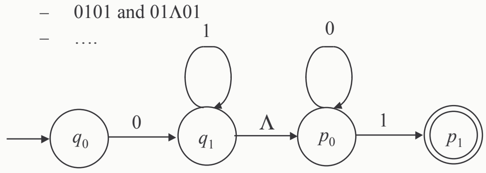
Concatenation

- $L_1 \cup L_2 = \mathbf{01^*0^*1}$
- But, what about this:



Corresponding strings

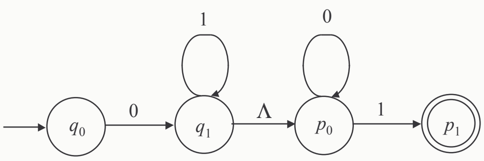
- But now… a much larger set of strings can be accepted: those with Λ symbols:
  - 01 and 0Λ1
  - 0101 and 01Λ01
  - ….



- The Λ's are never there in the input string: If the machine is an state with a Λ-transition, it can either:
  - Read the next symbol of Σ on the input string
  - Move (spontaneously) to the state after the Λ-transition!

Corresponding strings

- There is a string without Λ-transitions corresponding to a string with the Λ-transitions:
  - 01 corresponds to 0Λ1
  - 0101 corresponds to 01Λ01
- The sequence of transitions on the NFA-Λ will correspond to the strings with the Λ-transitions!



NFA-Λ

- Provides a further degree of expressive power
- Allows to express abstractions (specifications) in simple and direct ways!
- Provides the means to expressing the abstractions that can be expressed through Regular Expressions directly!

NFA with Λ-transitions: NFA-Λ

- A nondeterministic finite Automaton with Λ-transitions (NFA-Λ) is a 5-tuple:

  $(Q, \Sigma, q_0, A, \delta)$,

  where
  - Q is a finite set (of states)
  - Σ is a finite alphabet
  - $q_0 \in Q$ (the initial state)
  - $A \subseteq Q$ (the set of accepting states)
  - A transition function:

    $\delta: Q \times (\Sigma \cup \{\Lambda\}) \rightarrow 2^Q$

Transition function

- Transition function for DFA:

  $\delta: Q \times \Sigma \rightarrow Q$
- Transition function for NFA:

  $\delta: Q \times \Sigma \rightarrow 2^Q$
- Transition function for NFA-Λ

  $\delta: Q \times (\Sigma \cup \{\Lambda\}) \rightarrow 2^Q$
- We add the empty string to the set of input symbols

The extended transition function for NFA-Λ

- $\delta^*(q, x)$ denotes the set of states the machine is taken from the state q on the string x
- But now, a string $x \in \Sigma^*$ can correspond to many strings with Λ-transitions
- We need to make sure to reach the right set of states for x, regardless all Λ-transitions, in the paths from q to $\delta^*(q, x)$
- How can we include this last condition in the definition of δ^* for NFA-Λ?

Λ-closure of a set of states

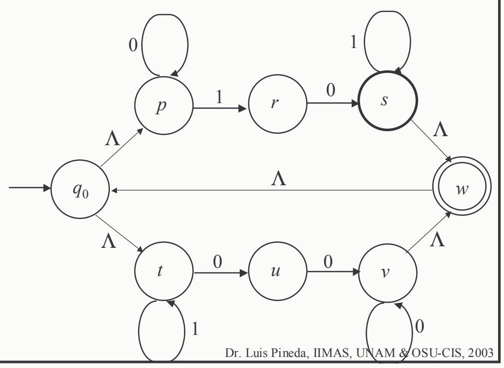
- Λ-Closure of a state (or a set of states): the set of states that can be reached through Λ-transitions from that state (or set of states)
- Let M $(Q, \Sigma, q_0, A, \delta)$ be an NFA-Λ, and S be any subset of Q. The Λ-Closure of S is the set $\Lambda(S)$ defined as follows:
  - Every element of S is an element of $\Lambda(S)$
  - For any $q \in \Lambda(S)$, every element of $\delta(q, \Lambda)$ is in $\Lambda(S)$
  - No other elements of Q are in $\Lambda(S)$

Computing the Λ-closure Λ(S) of S

- Let S be a set of states of a Λ-NFA
- Let $T = S$ where T is the states in the $\Lambda(S)$
  - For all $q \in T$ add to T all $\delta(q, \Lambda)$ which are not already in T
  - Repeat until there are no more states to be added!

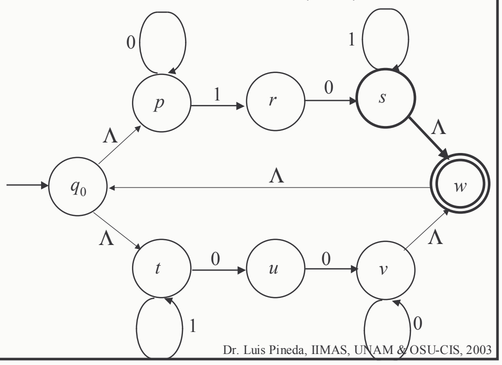
Computing Λ({s}):

$T = \{s\}$



Computing Λ({s}):

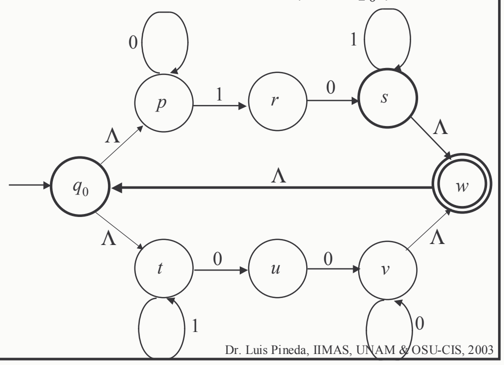
$\delta(s, \Lambda) = \{w\}$
$T = \{s, w\}$



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Computing Λ({s}):

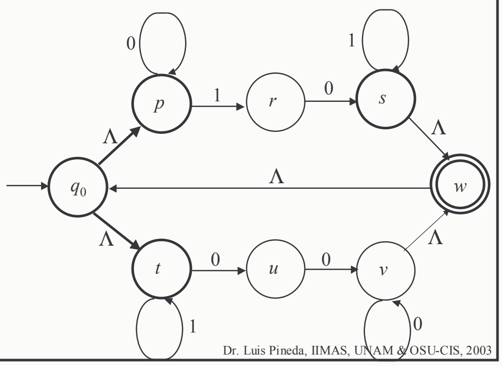
$\delta(w, \Lambda) = \{q_0\}$
$T = \{s, w, q_0\}$



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Computing Λ({s}):

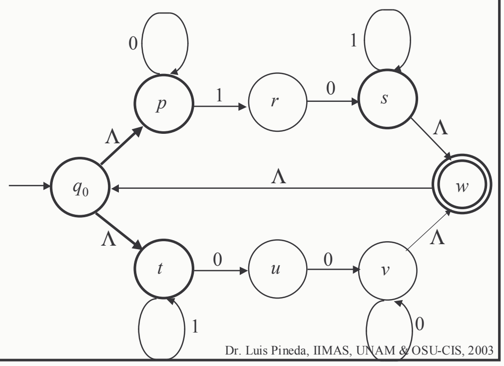
$\delta(q_0, \Lambda) = \{p, t\}$
$T = \{s, w, q_0, p, t\}$



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Computing Λ({s}):

$\delta(p, \Lambda) = \delta(t, \Lambda) = \phi$
$T = \{s, w, q_0, p, t\}$



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Λ-closure of S

- $\Lambda(S)$ computes the extended transition function δ^* for paths starting in the states of S that have Λ-transitions!
- If $\delta^*(q, y)$ is the set of all states that can be reached from q on y including Λ-transitions, then the set of states that can be reached from this set in one more transition on the symbol a is:

$$\bigcup_{r \in \delta^*(q,y)} \delta(r,a)$$

- The Λ-closure of this set includes any additional states that can be reached with Λ-transition

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Extended Transition function for NFA-Λ

- Let $M = (Q, \Sigma, q_0, A, \delta)$ be a NFA-Λ
- The extended transition function:

  $\delta^*: Q \times \Sigma^* \to 2^Q$

  is defined as follows:
  - For any $q \in Q$, $\delta^*(q, \Lambda) = \Lambda(\{q\})$.
  - For any $q \in Q$, $y \in \Sigma^*$ and $a \in \Sigma$:

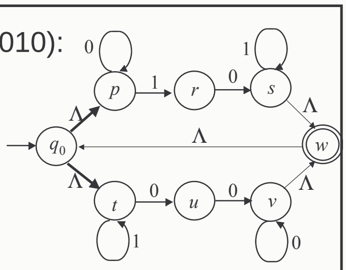
$$\delta^*(q, ya) = \Lambda\left(\bigcup_{r \in \delta^*(q,y)} \delta(r,a) \right)$$

  - A string x is accepted by M if $\delta^*(q_0, x) \cap A \neq \emptyset$
  - The language recognized by M is the set $\boldsymbol{L}(M)$ of all strings accepted by M

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

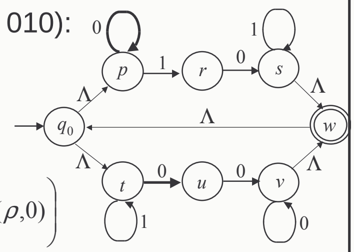
Computing: $\delta^*(q_0, 010)$:

$\delta^*(q_0, \Lambda) = \Lambda(\{q_0\})$
$= \{q_0, p, t\}$



Computing: $\delta^*(q_0, 010)$:

$\delta^*(q_0, \Lambda) = \{q_0, p, t\}$

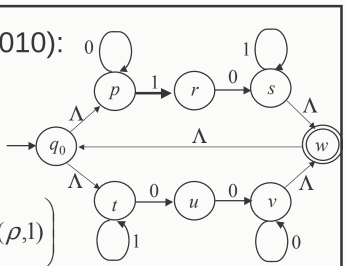


$$\delta^*(q_0, 0) = \Lambda\left(\bigcup_{r \in \delta^*(q_0, \Lambda)} \delta(\rho, 0)\right)$$

$= \Lambda(\delta(q_0, 0) \cup \delta(p, 0) \cup \delta(t, 0))$
$= \Lambda(\phi \cup \{p\} \cup \{u\})$
$= \Lambda(\{p, u\})$
$= \{p, u\}$

Computing: $\delta^*(q_0, 010)$:

$\delta^*(q_0, 0) = \{p, u\}$

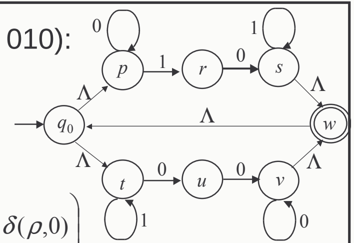


$$\delta^*(q_0, 01) = \Lambda\left(\bigcup_{r \in \delta^*(q_0, 0)} \delta(\rho, 1)\right)$$

$= \Lambda(\delta(p, 1) \cup \delta(u, 1))$
$= \Lambda(\{r\} \cup \phi)$
$= \Lambda(\{r\})$
$= \{r\}$

Computing: $\delta^*(q_0, 010)$:

$\delta^*(q_0, 01) = \{r\}$



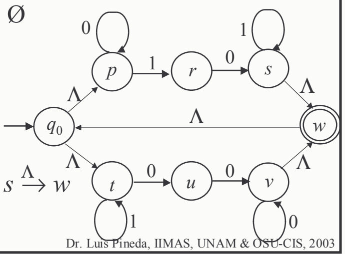
$$\delta^*(q_0, 010) = \Lambda\left(\bigcup_{r \in \delta^*(q_0, 01)} \delta(\rho, 0)\right)$$

$= \Lambda(\delta(r, 0))$
$= \Lambda(\{s\})$
$= \{s, w, q_0, p, t\}$

Acceptance by a NFA-Λ

- A string x is accepted by M if $\delta^*(q_0, x) \cap A \neq \emptyset$
- $A = \{w\}$
- $\delta^*(q_0, 010) = \{s, w, q_0, p, t\}$
- $\delta^*(q_0, x) \cap A = \{w\} \neq \emptyset$
- Accepting 010:

$$q_0 \xrightarrow{\Lambda} p \xrightarrow{0} p \xrightarrow{1} r \xrightarrow{0} s \xrightarrow{\Lambda} w$$



NFA-Λ are equivalent to NFA

- If $L \subseteq \Sigma^*$ is a language that is accepted by the NFA-Λ

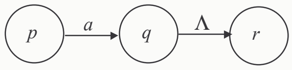
  $M = (Q, \Sigma, q_0, A, \delta)$

  then there is a NFA

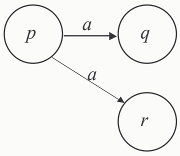
  $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$

  that also accepts L
- M_1 is the NFA without Λ-transitions corresponding to the NFA-Λ M

Eliminating Λ-transitions



is equivalent to



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Eliminating Λ-transitions

- Through δ^* will are able all the states that are reachable from any state of NFA-Λ (including Λ-transitions) on a string of any length
- These will include all the states of the NFA-Λ that can be reached on strings of length one!
- These strings are the symbols of the alphabet!
- We can bypass the Λ-transitions, substituting such transitions by the corresponding transitions on the symbol of the alphabet

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Eliminating Λ-transitions

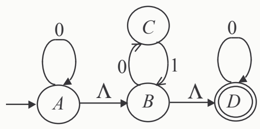
- To find the NFA corresponding to a NFA-Λ:
  - Giving the transition function for the NFA-Λ
  - Compute the extended transition function of the NFA-Λ for all the states for all the symbols of the alphabet (strings of length 1). This is, compute for every q_i

    $\delta^*(q_i, \Lambda) = \Lambda(\{q_i\})$ and

    $\delta^*(q_i, \Lambda a)$ for every $a \in \Sigma$
  - This provides the transition function for the corresponding NFA!

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

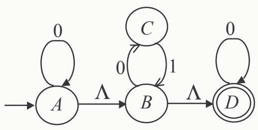
Converting NFA-Λ to NFA



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Converting NFA-Λ to NFA

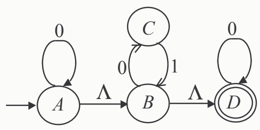
q	$\delta(q, \Lambda)$	$\delta(q, 0)$	$\delta(q, 1)$
A	$\{B\}$	$\{A\}$	ϕ
B	$\{D\}$	$\{C\}$	ϕ
C	ϕ	ϕ	$\{B\}$
D	ϕ	$\{D\}$	ϕ



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Converting NFA-Λ to NFA

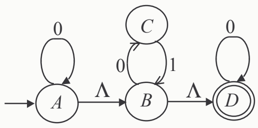
q	$\delta(q, \Lambda)$	$\delta(q, 0)$	$\delta(q, 1)$	$\delta^*(q, 0)$	$\delta^*(q, 1)$
A	$\{B\}$	$\{A\}$	ϕ		
B	$\{D\}$	$\{C\}$	ϕ		
C	ϕ	ϕ	$\{B\}$		
D	ϕ	$\{D\}$	ϕ		



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Converting NFA-Λ to NFA

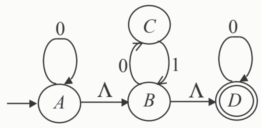
q	$\delta(q, \Lambda)$	$\delta(q, 0)$	$\delta(q, 1)$	$\delta^*(q, 0)$	$\delta^*(q, 1)$
A	$\{B\}$	$\{A\}$	ϕ	$\{A\}$	
B	$\{D\}$	$\{C\}$	ϕ		
C	ϕ	ϕ	$\{B\}$		
D	ϕ	$\{D\}$	ϕ		



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Converting NFA-Λ to NFA

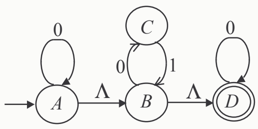
q	$\delta(q, \Lambda)$	$\delta(q, 0)$	$\delta(q, 1)$	$\delta^*(q, 0)$	$\delta^*(q, 1)$
A	$\{B\}$	$\{A\}$	ϕ	$\{A, B\}$	
B	$\{D\}$	$\{C\}$	ϕ		
C	ϕ	ϕ	$\{B\}$		
D	ϕ	$\{D\}$	ϕ		



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Converting NFA-Λ to NFA

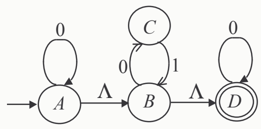
q	$\delta(q, \Lambda)$	$\delta(q, 0)$	$\delta(q, 1)$	$\delta^*(q, 0)$	$\delta^*(q, 1)$
A	$\{B\}$	$\{A\}$	ϕ	$\{A, B, C\}$	
B	$\{D\}$	$\{C\}$	ϕ		
C	ϕ	ϕ	$\{B\}$		
D	ϕ	$\{D\}$	ϕ		



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Converting NFA-Λ to NFA

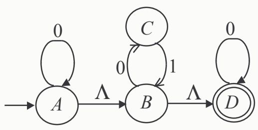
q	$\delta(q, \Lambda)$	$\delta(q, 0)$	$\delta(q, 1)$	$\delta^*(q, 0)$	$\delta^*(q, 1)$
A	$\{B\}$	$\{A\}$	ϕ	$\{A, B, C, D\}$	
B	$\{D\}$	$\{C\}$	ϕ		
C	ϕ	ϕ	$\{B\}$		
D	ϕ	$\{D\}$	ϕ		



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Converting NFA-Λ to NFA

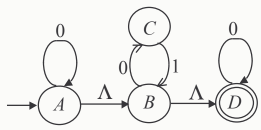
q	$\delta(q, \Lambda)$	$\delta(q, 0)$	$\delta(q, 1)$	$\delta^*(q, 0)$	$\delta^*(q, 1)$
A	$\{B\}$	$\{A\}$	ϕ	$\{A, B, C, D\}$	
B	$\{D\}$	$\{C\}$	ϕ		
C	ϕ	ϕ	$\{B\}$		
D	ϕ	$\{D\}$	ϕ		



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Converting NFA-Λ to NFA

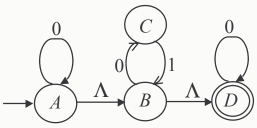
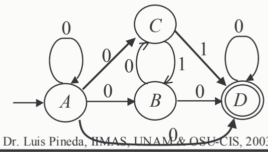
q	$\delta(q, \Lambda)$	$\delta(q, 0)$	$\delta(q, 1)$	$\delta^*(q, 0)$	$\delta^*(q, 1)$
A	$\{B\}$	$\{A\}$	ϕ	$\{A, B, C, D\}$	ϕ
B	$\{D\}$	$\{C\}$	ϕ	$\{C, D\}$	ϕ
C	ϕ	ϕ	$\{B\}$	ϕ	$\{B, D\}$
D	ϕ	$\{D\}$	ϕ	$\{D\}$	ϕ



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

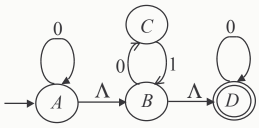
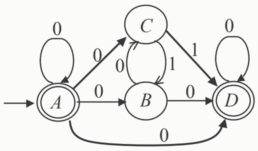
Converting NFA-Λ to NFA

q	$\delta(q, \Lambda)$	$\delta(q, 0)$	$\delta(q, 1)$	$\delta^*(q, 0)$	$\delta^*(q, 1)$
A	$\{B\}$	$\{A\}$	ϕ	$\{A, B, C, D\}$	ϕ
B	$\{D\}$	$\{C\}$	ϕ	$\{C, D\}$	ϕ
C	ϕ	ϕ	$\{B\}$	ϕ	$\{B, D\}$
D	ϕ	{D}	ϕ	$\{D\}$	ϕ

Converting NFA-Λ to NFA

Finally, if there are accepting states that can be reached by Λ-transition from the initial state of the NFA- Λ, then the initial state of the corresponding NFA is also accepting

NFA-Λ are equivalent to NFA

- If $L \subseteq \Sigma^*$ is a language that is accepted by the NFA-Λ $M = (Q, \Sigma, q_0, A, \delta)$ then there is a NFA $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ that also accepts L
- This is the case if and only if:

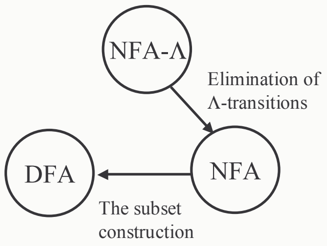
$$\delta_1^*(q, y) = \delta^*(q, y)$$

- $Q_1 = Q$ and the accepting states:

$$A_1 = \begin{cases} A \cup \{q_0\} & \text{if } \Lambda(\{q_0\}) \cap A \neq \Phi \text{ in } M \\ A & \text{otherwise} \end{cases}$$

- The proof is by induction

The Family of FA



Converting FA into NFA-Λ

- We have so far:
  - L can be recognized by a FA
  - L can be recognized by a NFA
  - L can be recognized by a NFA-Λ
- These three statements are equivalent:
  - For any NFA there is an equivalent FA (the subset construction)
  - For any NFA-Λ there is an equivalent NFA (eliminating Λ-transitions)
  - For any FA there is an equivalent NFA-Λ!

FA are equivalent to NFA-Λ

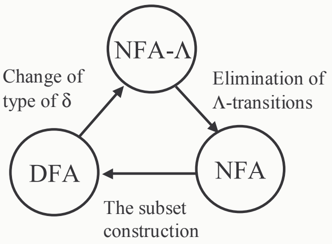
- Suppose that L is accepted by a DFA $M = (Q, \Sigma, q_0, A, \delta)$; we construct an NFA-Λ $M_1 = (Q, \Sigma, q_0, A, \delta_1)$
  - The same set of states
  - The same alphabet
  - The same initial state
  - The same set of accepting states!
  - We define δ_1 of type:

    $\delta_1 : Q \times (\Sigma \cup \{\Lambda\}) \rightarrow 2^Q$

    as follows (for any $q \in Q$ and $a \in \Sigma$):

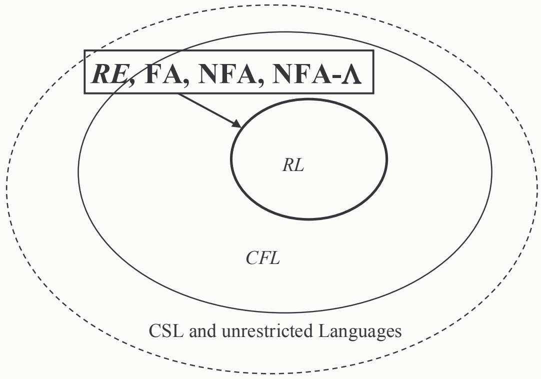
    $\delta(q, \Lambda) = \phi \qquad \delta(q, a) = \{\delta(q, a)\}$
- The trivial NFA-Λ with no non-deterministic or Λ-transitions!

The Family of FA



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

The 4th description of *RL*



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003