CS442/542

Lexical Analysis
Part 2

Example Regular Expressions over the alphabet {0, 1}

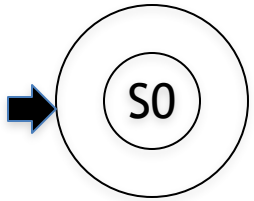
Regular Expression	Language
• 0	• { 0 }
• 0 | 1	• {0, 1}
• 0 (0 | 1)	• {00, 01}
• 1*	• { x | x is a string of 0 or more 1s}
• (0 | 1) *	• { x | x is any string of 0s and 1s including the empty string}

RE -> NFA

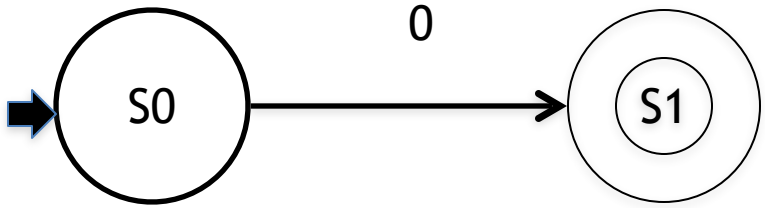
Regular Expression

NFA

ϵ



0

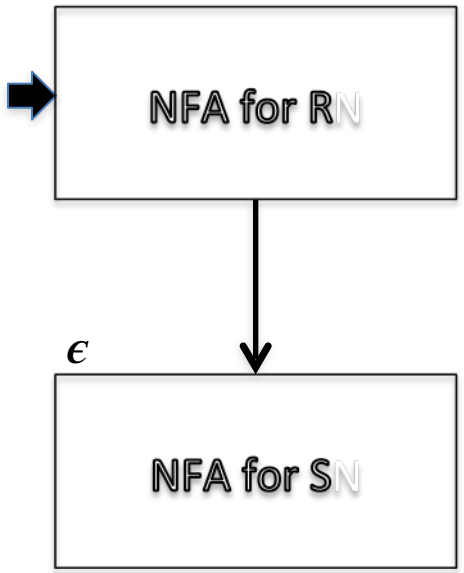


RE -> NFA

Regular Expression (Assume R and S are regular expressions)

NFA

R S



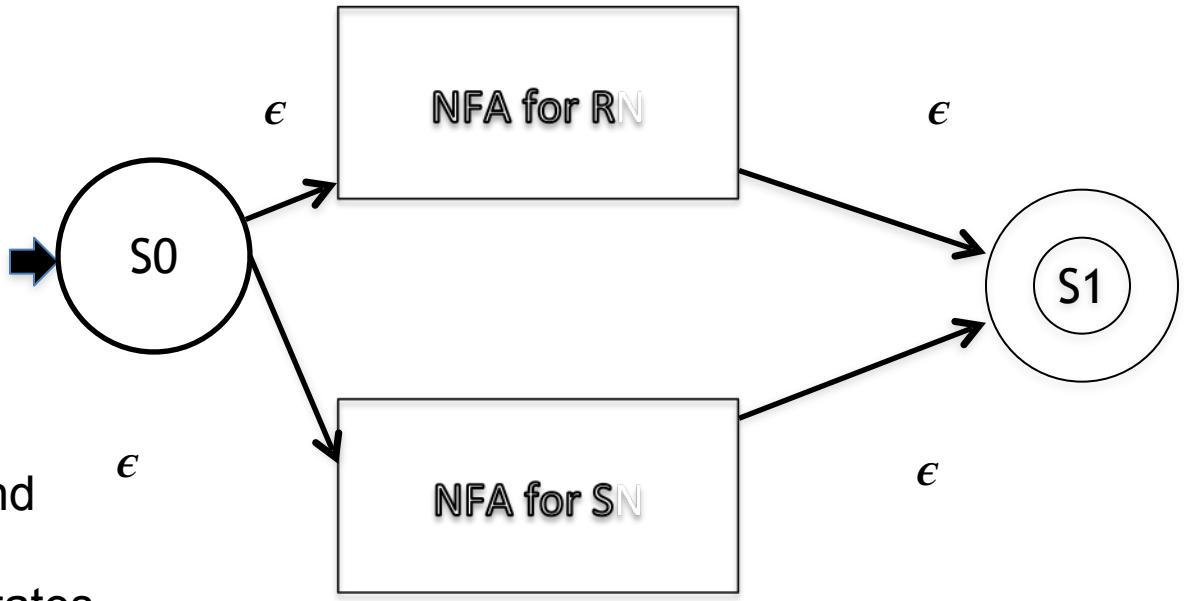
The start state of the NFA for R is the start state of the new machine. The final state in the NFA for R connects via an epsilon transition to the start state of the NFA for S. The final state in the NFA for S is the final state for the new machine.

RE -> NFA

Regular Expression (Assume R and S are regular expressions)

R | S

NFA



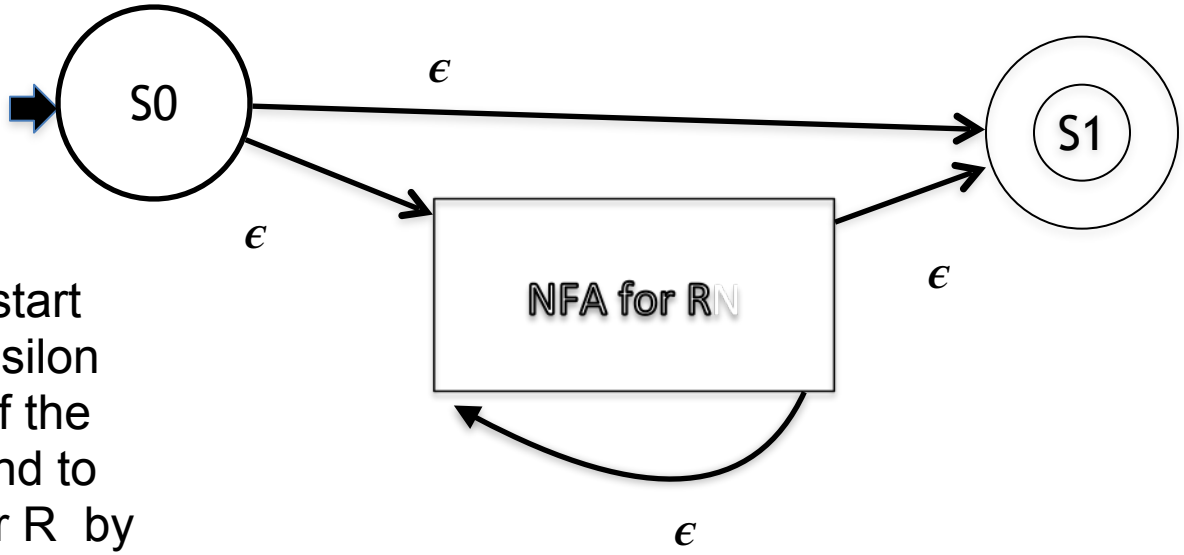
S0 connects to the start states of the NFA for R and the NFA for S by epsilon transitions and the final states of the NFA for R and NFA for S connect to S1 by epsilon transitions

RE -> NFA

Regular Expression (Assume R is a regular expressions)

NFA

R*



S0 connects to S1 and the start state of the NFA for R by epsilon transitions. The final state of the NFA for R connects to S1 and to the start state of the NFA for R by epsilon transitions.

Example Problems

- Construct NFAs for the following regular expressions.
  - 0
  - 00
  - 0 | 1
  - (0 | 1) *
  - 0*(10*10*)*