Formal Languages and Automata Theory(18IS54)

Unit 2

Regular Expressions and Languages

- Regular expressions are closely related to NFA and can be thought of as a userfriendly alternative to NFA notation.
- Regular expressions serve as an input langauge for many system that process strings.
- Operators of regular expressions
  - ✔ Union of two languages L and M denoted by L U M.
    L={001,10,111} and M = { ε,001} LUM={ε,10,001,111}
  - ✔ Concatenation of languages L and M
    LM={001,10,111,001001,10001,111001}
  - ✔ CLosure of a language denoted by L* is set of those strings that can be formed taking any number of strings from L, possibly with repetitions and concatenating all of them.
    L^2 = {00,011,110,1111} fro L={0,11}

Building regular expressions

- Regular expressions E describe the langauge it represents which is denoted by L(E)

Regular Expressions and Languages

Building regular expressions

- Regular expressions E describe the langauge it represents which is denoted by L(E)

Basis: It consists of 3 parts

1. The constants ε and Ø are regular expressions denoting the langauge {ε} and Ø respectively.
2. If a is any symbol then a is regular expression.
3. A variable usually capitalised and italic such as L is a variable representing any langauge.

Induction:

1. If E and F are regular expressions then E+F is a regular expression denoting the union of L(E) and L(F).
2. If E and F are regular expressions then EF is a regular expressoins denoting the concatenation of L(E) and L(F).
3. If E is aregular expression then E* is a regular expression denoting closure of L(E)
4. If E is a regular expression then (E) a parameterised E is also a regular expression denoting same langauge as E.

Finite Automata and Regular Expressions

Theorem: If L=L(A) for some DFA A, then there is a regular expression such that L = L(R)

Proof :

Basis: for k=0 two kinds of paths

An arc from node i to node j

A path of length 0 that consists of only some node i if i != j then only case 1 is possible

Examine DFA A

Induction:

Suppose a path goes from i to j that goes through the state higher then k

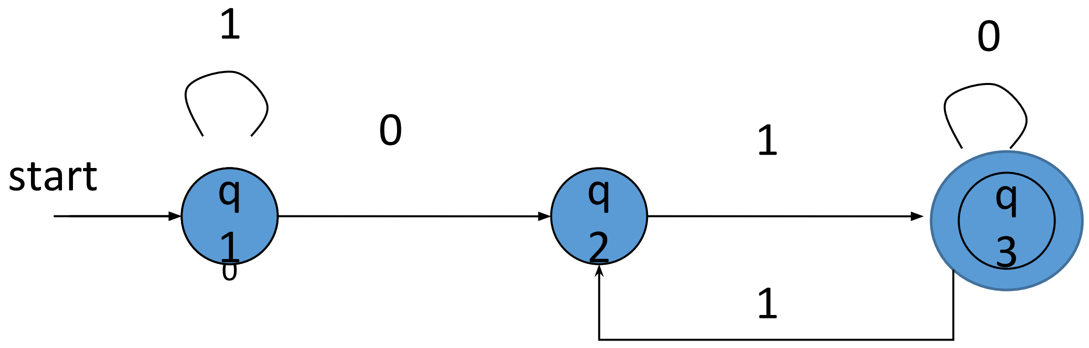
Two possible cases

1. path does not go through k at all represented by $R_{ij}^{(k-1)}$.
2. path goes through state k atleast once

$R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$

$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$

1*01(0+11)*



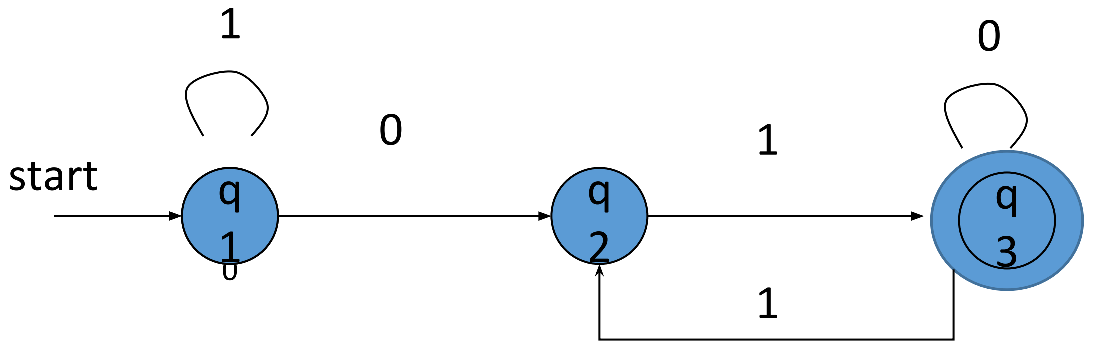
Finite Automata and Regular Expressions

1*01(0+11)*
no states

$R_{11}^{(0)} = 1+\varepsilon$
$R_{12}^{(0)} = 0$
$R_{13}^{(0)} = \emptyset$
$R_{21}^{(0)} = \emptyset$
$R_{22}^{(0)} = \varepsilon$
$R_{23}^{(0)} = 1$
$R_{31}^{(0)} = \emptyset$
$R_{32}^{(0)} = 1$
$R_{33}^{(0)} = 0+\varepsilon$

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$$

$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)} (R_{11}^{(0)})^* R_{12}^{(0)}$
$=0 + (1+\varepsilon)(1+\varepsilon)^* 0 = 1^*0$
$R_{11}^{(1)} = 1^*$
$R_{33}^{(3)}$



Finite Automata and Regular Expressions

1*01(0+11)*

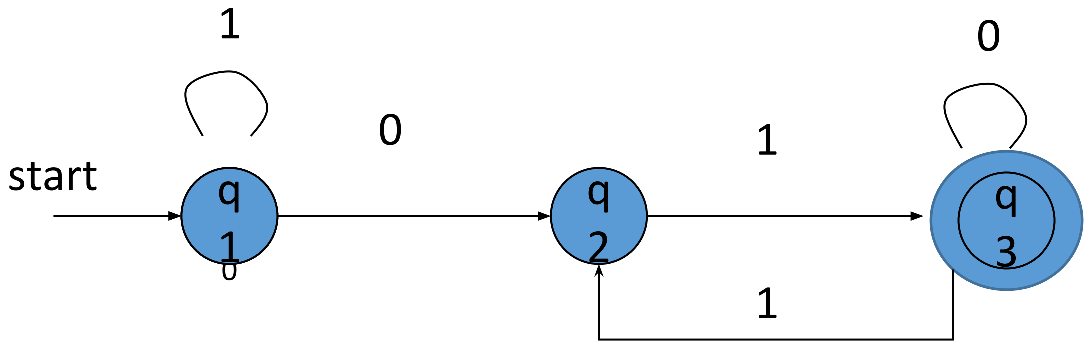
no states

$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$

state 1 only

$R_{11}^{(0)} = 1+\varepsilon$
$R_{12}^{(0)} = 0$
$R_{13}^{(0)} = \emptyset$
$R_{21}^{(0)} = \emptyset$
$R_{22}^{(0)} = \varepsilon$
$R_{23}^{(0)} = 1$
$R_{31}^{(0)} = \emptyset$
$R_{31}^{(0)} = 1$
$R_{33}^{(0)} = 0+\varepsilon$

$R_{11}^{(1)} = 1^*$
$R_{12}^{(1)} = 1^*0$
$R_{13}^{(1)} = \emptyset$
$R_{21}^{(1)} = \emptyset$
$R_{22}^{(1)} = \varepsilon$
$R_{23}^{(1)} = 1$
$R_{31}^{(1)} = \emptyset$
$R_{31}^{(1)} = 1$
$R_{33}^{(1)} = 0+\varepsilon$



Finite Automata and Regular Expressions

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)}) \, (R_{kk}^{(k-1)})^* \, R_{kj}^{(k-1)}$$

1*01(0+11)*

no states

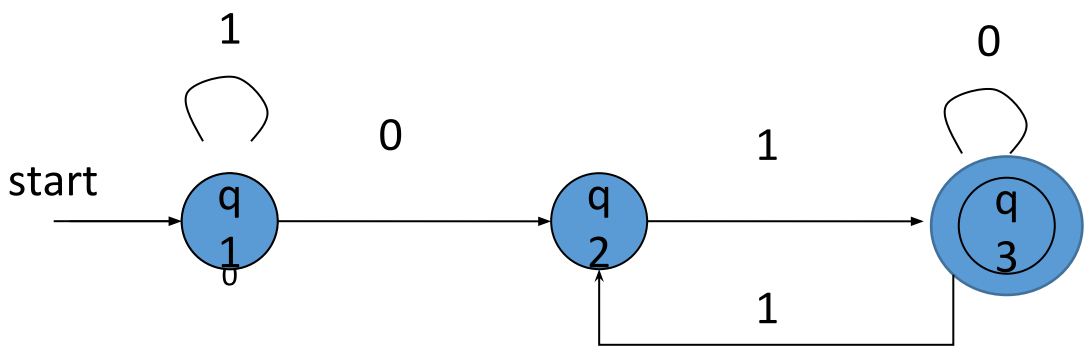
$R_{11}^{(0)} = 1+\varepsilon$
$R_{12}^{(0)} = 0$
$R_{13}^{(0)} = \emptyset$
$R_{21}^{(0)} = \emptyset$
$R_{22}^{(0)} = \varepsilon$
$R_{23}^{(0)} = 1$
$R_{31}^{(0)} = \emptyset$
$R_{31}^{(0)} = 1$
$R_{33}^{(0)} = 0+\varepsilon$

state 1 only

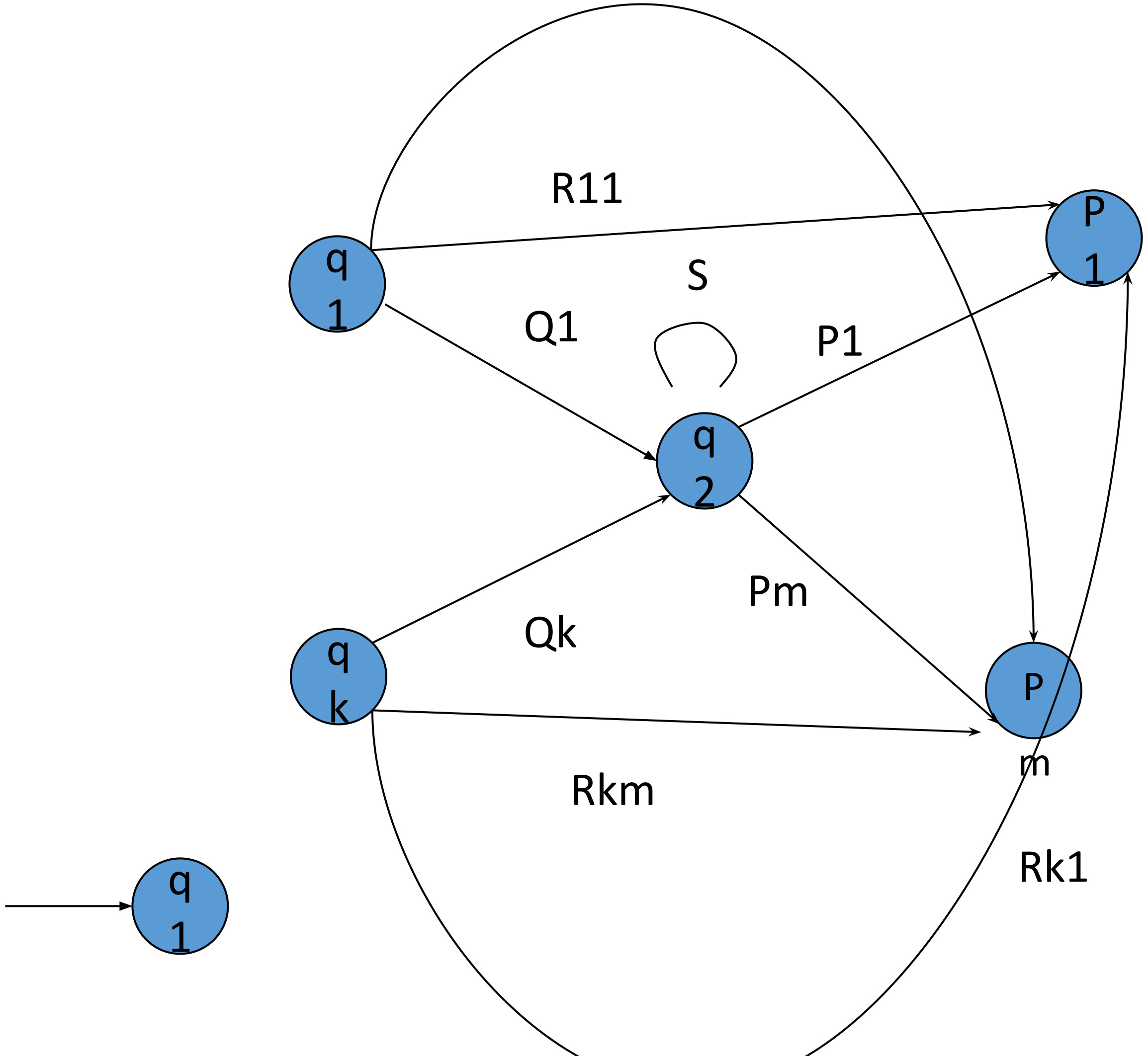
$R_{11}^{(1)} = 1^*$
$R_{12}^{(1)} = 1^*0$
$R_{13}^{(1)} = \emptyset$
$R_{21}^{(1)} = \emptyset$
$R_{22}^{(1)} = \varepsilon$
$R_{23}^{(1)} = 1$
$R_{31}^{(1)} = \emptyset$
$R_{31}^{(1)} = 1$
$R_{33}^{(1)} = 0+\varepsilon$

state 1 and 2 only

$R_{11}^{(2)} = 1^*$
$R_{12}^{(2)} = 1^*0$
$R_{13}^{(2)} = 1^*01$
$R_{21}^{(2)} = \emptyset$
$R_{22}^{(2)} = \varepsilon$
$R_{23}^{(2)} = 1$
$R_{31}^{(2)} = \emptyset$
$R_{31}^{(2)} = 1$
$R_{33}^{(2)} = 0+\varepsilon +11$



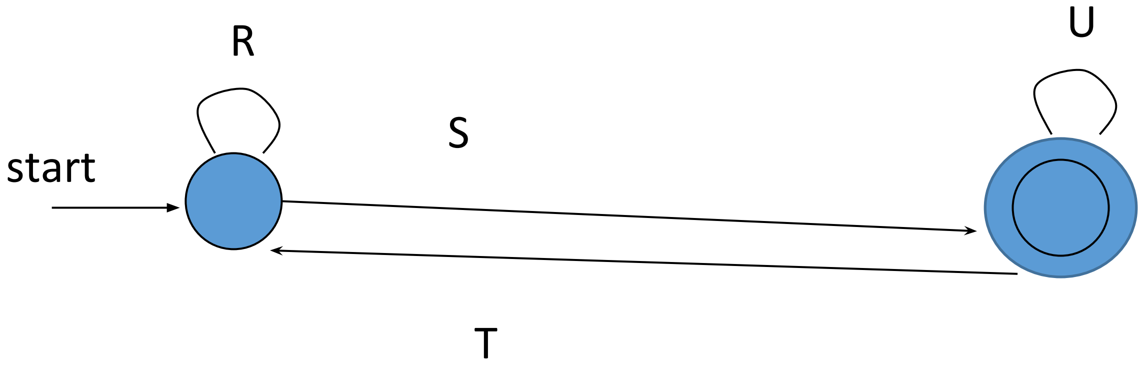
Converting DFA to Regular Expressions by eliminating states

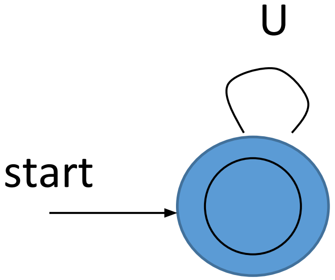


R11 + Q1S*P1

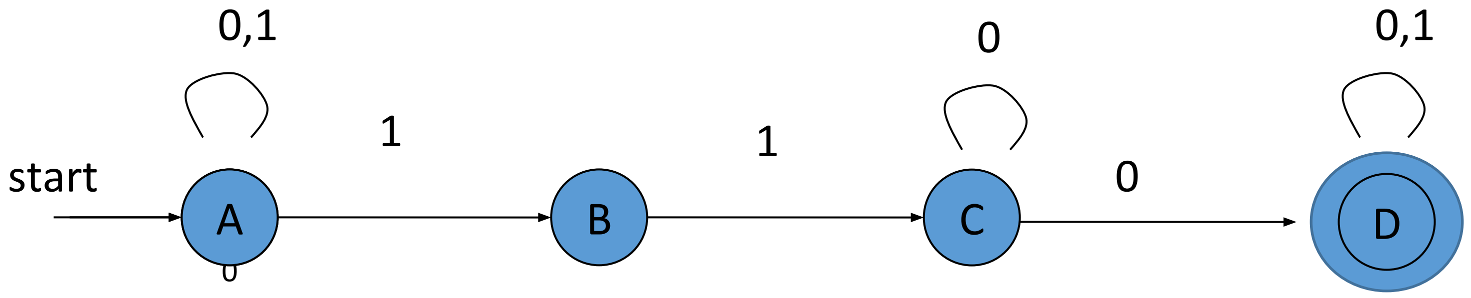
R1m+Q1S*Pm

RK1+QKS*P1

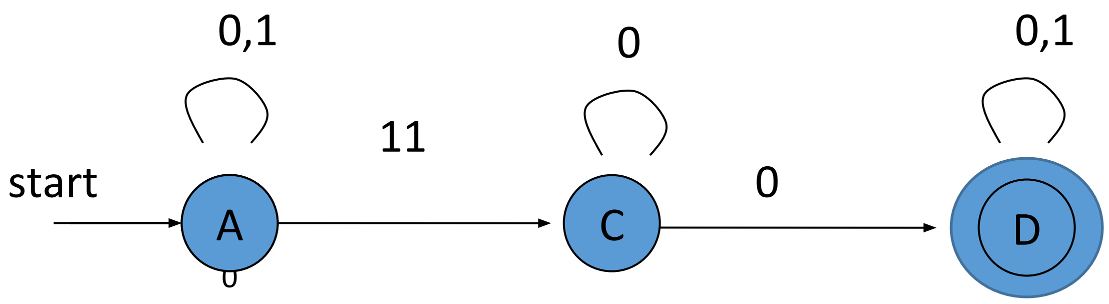




Design a NFA which accepts all strings containing 110 convert it to regular expression

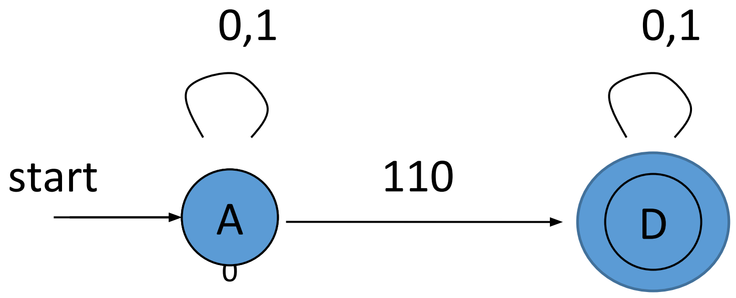


Design a NFA which accepts all strings containing 110 convert it to regular expression



Design a NFA which accepts all strings containing 110 convert it to regular expression

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$$



R + (S U * T)* S U*

R=0+1, S=110, T= Ø, U=0+1 SU*T = Ø

(1+0)* 110 (1+0)*