



PES UNIVERSITY, Bangalore
(Established under Karnataka Act No. 16 of 2013)
Department of Computer Science & Engineering

Automata Formal Languages & Logic

Q&A - Regular Expression(RE)

1) Construct regular expression for each of the following

- a) Binary strings with at least two occurrences of at least two consecutive 1 s, the two occurrences not being adjacent (i.e., 011011 is acceptable but 011111 is not).

Solution: There must be at least one 0 between the two occurrences of 11:

$$(0+1)^*11(0+1)^*0(0+1)^*11(0+1)^*$$

- b) Binary strings containing at least one 00 and at least one 11.

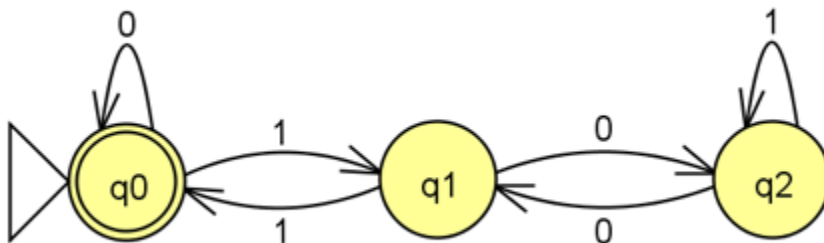
Solution: 00 can come first and then 11 or 11 first and then 00:

$$(0+1)^*00(0+1)^*11(0+1)^* + (0+1)^*11(0+1)^*00(0+1)^*$$

2)convert the following RegEx to an equivalent NFA/DFA

- a) $(0+\lambda)(1+\lambda)(1+2)^*0(2+1)^*$

Binary strings representing positive integers divisible by 3.

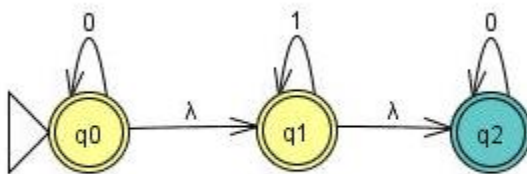


PES UNIVERSITY, Bangalore
(Established under Karnataka Act No. 16 of 2013)
Department of Computer Science & Engineering

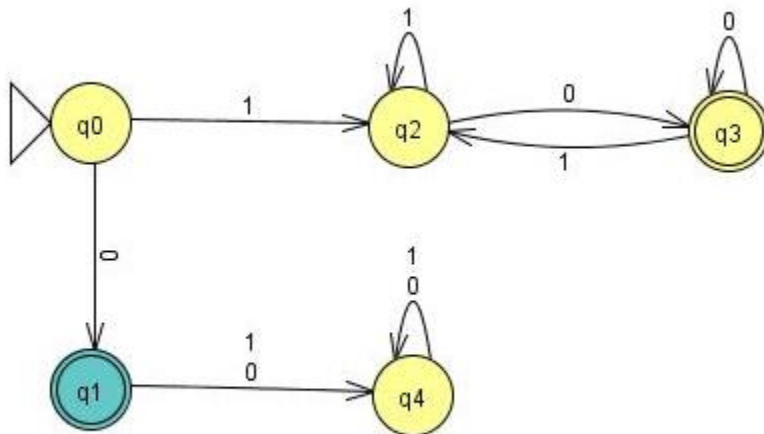
Automata Formal Languages & Logic

b) $0^*1^*0^*$ with three states

Solution:



3) Convert the following DFA/NFA to an equivalent RegEx



Solution:

$$0 + 11^*0(0 + 11^*0)^*$$



PES UNIVERSITY, Bangalore
(Established under Karnataka Act No. 16 of 2013)
Department of Computer Science & Engineering

Automata Formal Languages & Logic

4) Are the following pairs of RegEx's equivalent (do they represent same set of strings)
 $(0+\lambda)(11^*0)^*(1+\lambda)$ and $(1+\lambda)(011^*)^*(0+\lambda)$

Solution: Yes, both can start with 0 or 1; both can end with 0 or 1; both do not allow consecutive 0 s. Both represent the language of all binary strings without consecutive 0 s (except 11^* which neither of them include). The first RegEx excludes consecutive 0 s by requiring every 0 except the first one to be preceded by a 1; the second RegEx achieves the same result by requiring every 0 except the last one to be followed by a 1.