Theory of Computing

Finite Automata : NFA

TD 3

2ND YEAR - ENSIA

PRE-TUTORIAL EXERCISE

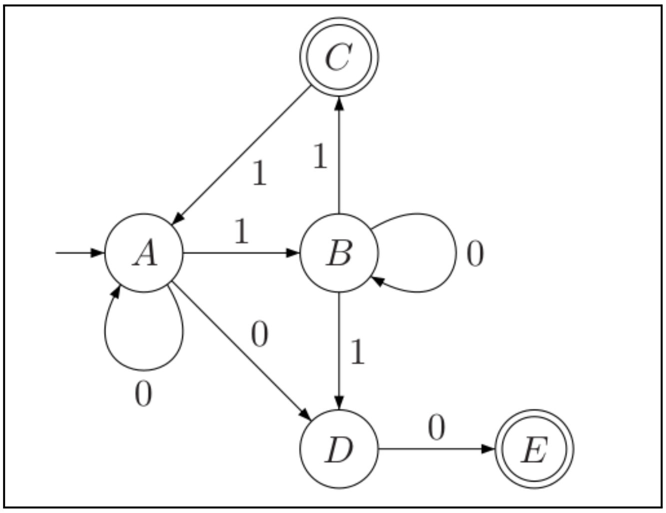
Draw the NFA for the language {w| w ends with 00} with **three** states

EXERCISES

Exercise C1 (Constructing NFA) :

Give state diagrams of NFAs with the specified number of states recognizing each of the following languages. In all parts, the alphabet is {0,1}.

1. The language {0} with **two** states
2. The language {w| w contains the substring 0101 (i.e., w = x0101y for some x and y)} with **five** states
3. The language {w| w contains an even number of 0s, or contains exactly two 1s} with **six** states
4. The language 0* with **one** state
5. The language 0* 1* 0+ with **three** states
6. **Let Σ = {a, b, c}** and let L = { w ∈ Σ* | some character in Σ appears at most twice in w }. Either a or b or c appear twice at most in a word. (Accepted : **aa**bbb, cccc**b**,ccc, Rejected words: aaabbbcccc, aaabbbccc).



a : means : ε , a,aa,aaa….*

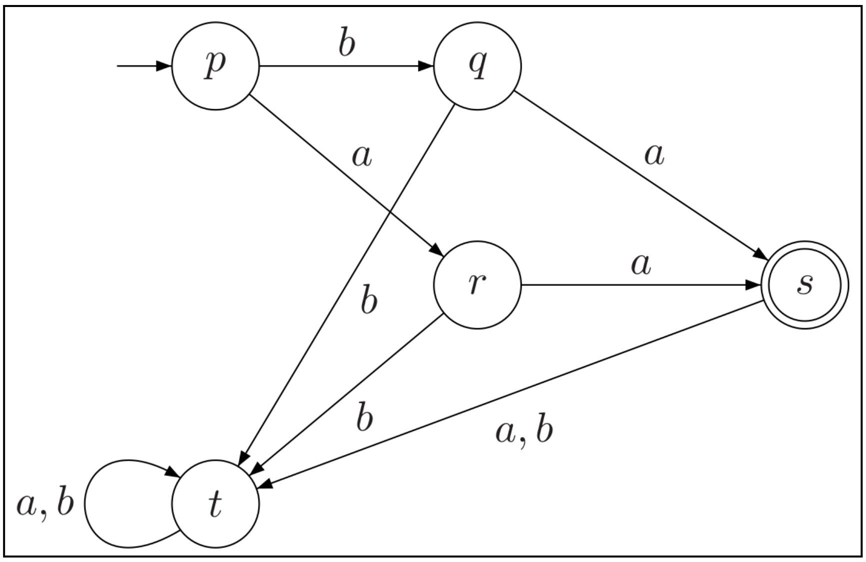
ab+ : means : ab , abb, abbb,abbb…

Exercise C2 (Converting NFA → DFA):

Convert the following NFA to their equivalent DFA:

Exercise C3 (Minimizing DFA) :

Minimize the following finite automaton :



Exercise P1 (Optional) :

Let Σ = {a, b, c}. Give an NFA for the language L containing all strings in Σ* which have an a or a c in the last four positions. E.g. bbabbb and abbbcb are both in L, but acabbbb is not. Notice that strings of length four or less are in L exactly when they contain an a or a c (NO MORE THAN 8 STATES)

Exercise P2 (Optional) :

1. Give an NFA recognizing the language (01 ∪ 001 ∪ 010)* .
2. Convert this NFA to an equivalent DFA. Give only the portion of the DFA that is reachable from the start state.

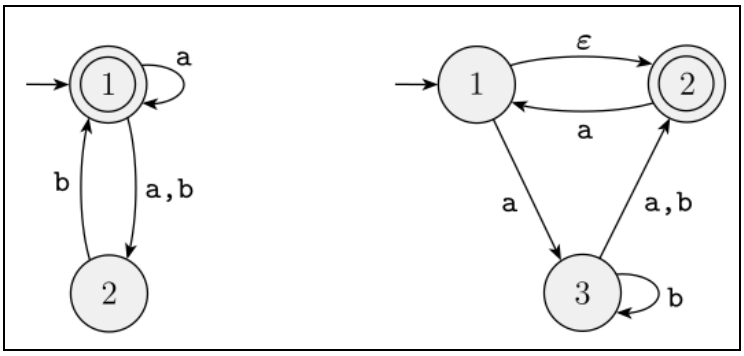
Exercise P3 (Optional):

Give state diagrams of NFAs with the specified number of states recognizing each of the following languages. In all parts, the alphabet is {0,1}.

a. The language 1* (001+)* with three states
b. The language {ε} with one state
c. The language of strings of odd length
d. The language of strings which contain an even number of 0’s.
e. The language of binary numbers which are divisible by 4.
f. All strings beginning and ending with abb
g. All strings containing abb or bab (or both) as a substring
h. All strings NOT containing abb as a substring
i. L = {w Ė Σ^* | w contains two 0s or exactly two 1s, and or is exclusive}.
j. L = {w | w are of the form $0^*1^*0^+$} (note: $0^+=0^*0=00^*$).

Exercise P4 (Optional) :

Convert the following two NFAs to its DFA equivalents :



Exercise P5 (Optional):

Let F be the language of all strings over {0,1} that do not contain a pair of 1s that are separated by an odd number of symbols. Give the state diagram of a DFA with five states that recognizes F . (You may find it helpful first to find a 4-state NFA for the complement of F .)

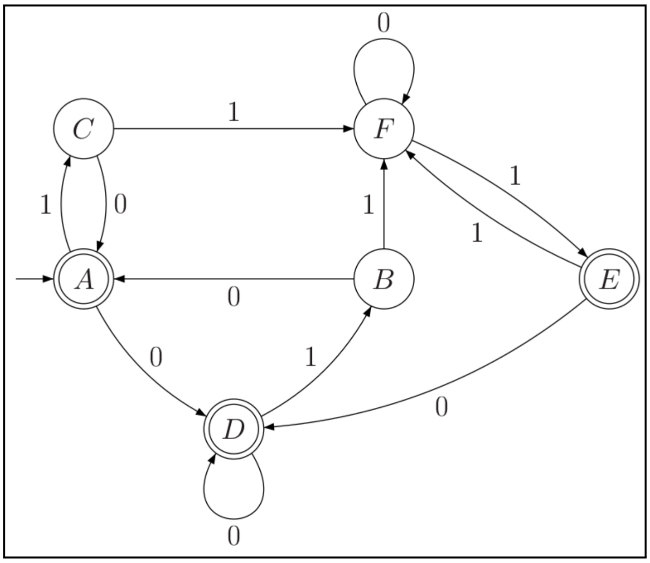
Exercise P6 (Optional) :

Let n be a positive integer and L = {x ∈ {a, b}* | |x| = n and $n_a(x)$ = $n_b(x)$}. What is the minimum number of states in any FA that accepts L? Give reasons for your answer.

Note that : $n_a(x)$ is the number of occurrences of a in the string x

Exercise P7 (Optional) :

Minimize the following finite automaton :

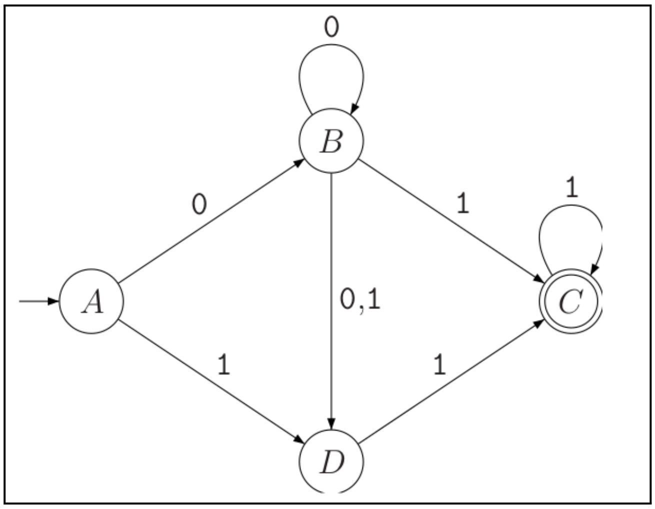


Exercise P8 (Optional) :

Show by giving an example that if M is an NFA that recognizes language C, swapping the accept and nonaccept states in M doesn’t necessarily yield a new NFA that recognizes the complement of C. Is the class of languages recognized by NFAs closed under complement? Explain your answer.

Exercise P9 (Optional) :

Convert the following nondeterministic finite automaton to equivalent deterministic finite automata.



Challenge 1 :

Construct the deterministic finite automata (DFA) for the following languages :

- L = { w ∈ {a, b, c}* | at least one of a, b, or c is not in w }
- L = { a^i b^j | i ≥ 0, j ≥ 0, i + j is an even number} | a^i means a being repeated i times aaaaa …..