Compiler

Report DFA and NFA

Gamal Mohamed El Sayed Mohamed

Number:37

NFA for the L+ of a regular expression:

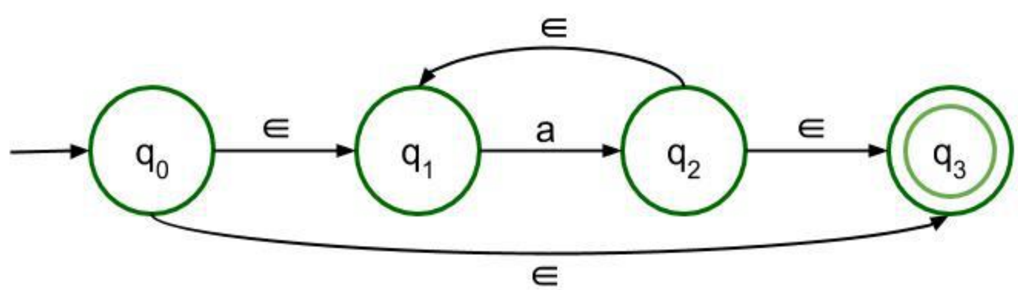
Non-Deterministic Finite Automata and ∈-Non-Deterministic Finite Automata are almost the same except for their transition function and there are a few special rules for construction of ∈-NFA

```
∈-NFA is defined in 5 tuple representation {Q, q0, Σ, δ, F} where –
Q is the set of all states,
q0 is the initial state,
Σ is the set of input symbols,
δ is the transition function which is δ:Q × (Σ∪∈)->2Q and
F is the set of final states.
```

∈-NFA for a+:

This structure is for a+ which means there must be at least one 'a' in the expression. It is preceded by epsilon and also succeeded by one. There is epsilon feedback from state q2 to q1 so that there can be more than one 'a' in the expression

∈-NFA for a*:



This structure is for a* which means there can be any number of 'a' in the expression, even 0. The previous structure is just modified a bit so that even if there is no input symbol, i.e. if the input symbol is null, then also the expression is valid

Minimization of DFA

Minimization of DFA means reducing the number of states from given FA. Thus, we get the FSM(finite state machine) with redundant states after minimizing the FSM.

We have to follow the various steps to minimize the DFA. These are as follows:

Step 1: Remove all the states that are unreachable from the initial state via any set of the transition of DFA.

Step 2: Draw the transition table for all pair of states.

Step 3: Now split the transition table into two tables T1 and T2. T1 contains all final states, and T2 contains non-final states.

Step 4: Find similar rows from T1 such that:

1. 1. δ (q, a) = p
2. 2. δ (r, a) = p

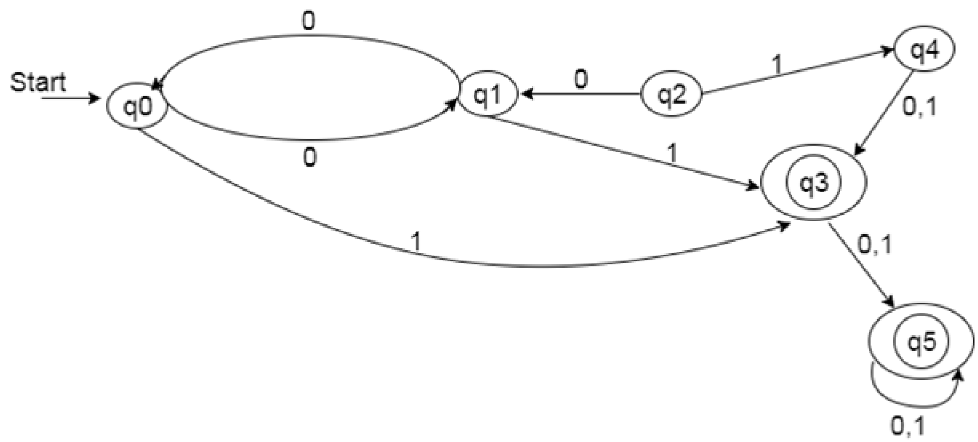
That means, find the two states which have the same value of a and b and remove one of them.

Step 5: Repeat step 3 until we find no similar rows available in the transition table T1.

Step 6: Repeat step 3 and step 4 for table T2 also.

Step 7: Now combine the reduced T1 and T2 tables. The combined transition table is the transition table of minimized DFA.

Example:



Solution:

Step 1: In the given DFA, q2 and q4 are the unreachable states so remove them.

Step 2: Draw the transition table for the rest of the states.

State	0	1
→q0	q1	q3
q1	q0	q3
*q3	q5	q5
*q5	q5	q5

Step 3: Now divide rows of transition table into two sets as:

1. One set contains those rows, which start from non-final states:

State	0	1
q0	q1	q3
q1	q0	q3

2. Another set contains those rows, which starts from final states.

State	0	1
q3	q5	q5

q5	q5	q5

Step 4: Set 1 has no similar rows so set 1 will be the same.

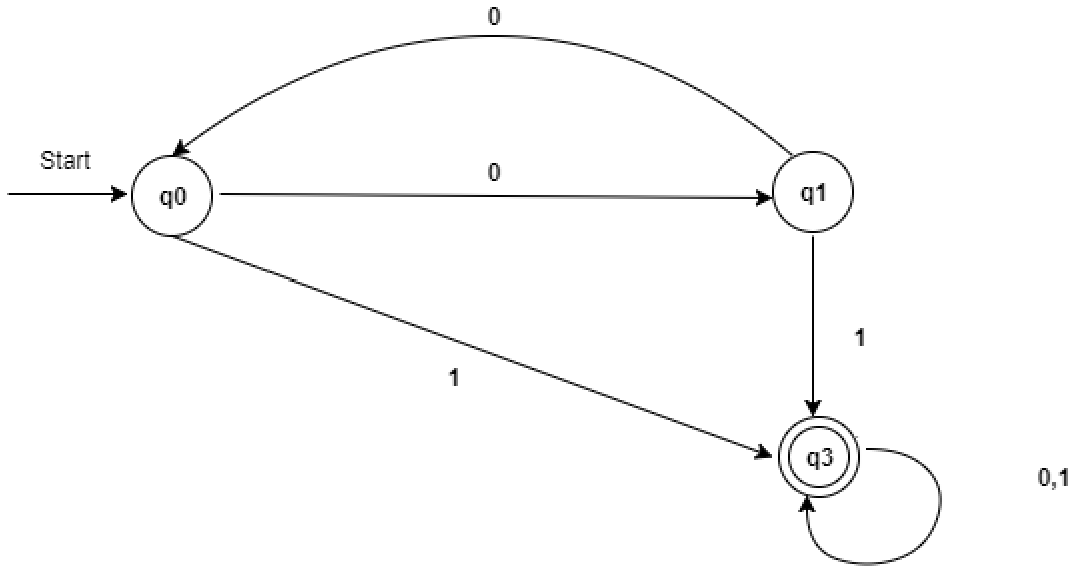
Step 5: In set 2, row 1 and row 2 are similar since q3 and q5 transit to the same state on 0 and 1. So skip q5 and then replace q5 by q3 in the rest.

State	0	1
q3	q3	q3

Step 6: Now combine set 1 and set 2 as:

State	0	1
→q0	q1	q3
q1	q0	q3
*q3	q3	q3

Now it is the transition table of minimized DFA.


0
Start
q0
0
q1
1
1
q3
0,1