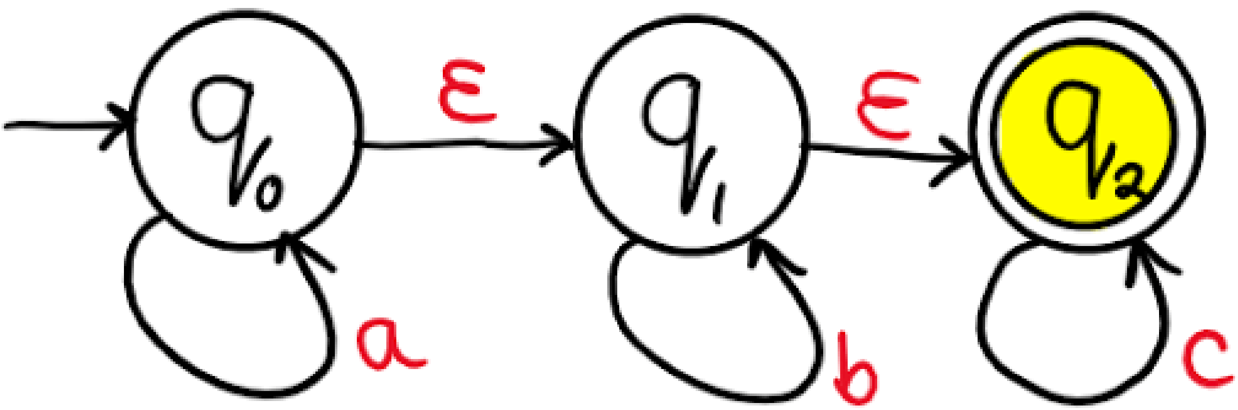
NFA with ∈ moves is exactly same as NFA without ∈ moves.

But differece exist in the transition function δ. δ must include information about ∈ transitions.

> *NFA with ∈-Moves has 6 tuples (Q, Σ, δ, q0, F).*
>
> *Where,*
>
> - *Q = finite set of states.*
> - *Σ = finite set input symbols.*
> - *δ = transition function that maps Q × (Σ ∪{∈}) to $2^Q$.*
> - *q0 = initial state.*
> - *F = set of final states.*

The non-deterministic finite automaton can be extended to include the transitions on null/empty input ∈.

## For example,

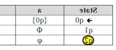



NFA with ∈

In this NFA with epsilon,

- It accept an input string ‘aabc’.
- Or string as number of a’s followed by number of b’s followed by number of c’s.
- The string ‘aabc’ is accepted by the NFA by following the path with labels a, a, ∈, b, ∈, c.

## Transition table for above NFA.



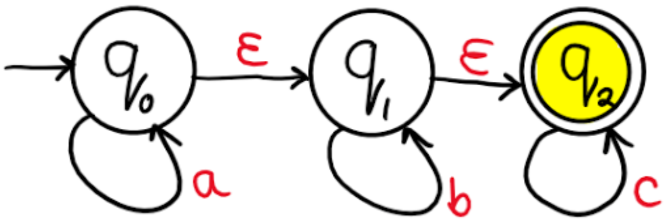
| State | a |
|---|---|
| → q0 | {q0} |
| q1 | Φ |
| q2 | φ |

### ∈-closure

∈-closure of a state q is a set of states following by all transitions of q that are labeled as ∈.

- ∈-closure (q0) = (q0, q1, q2)
- ∈-closure (q1) = (q1, q2)
- ∈-closure (q2) = (q2)

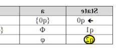
## NFA with ∈ to NFA without ∈




NFA with ∈

Transition diagram

Transition table NFA with ∈

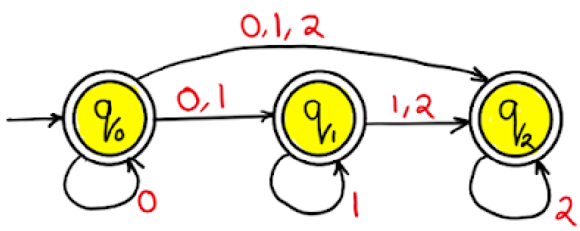


First find out ∈ closure:∈-closure

- (q0) = (q0, q1, q2)
- ∈-closure (q1) = (q1, q2)
- ∈-closure (q2) = (q2)

Transition table NFA without ∈

| State | a | b | c |
|---|---|---|---|
| →q0 | {q0, q1, q2} | {q1, q2} | {q2} |
| q1 | Φ | {q1, q2} | {q2} |
| q2 | ϕ | ϕ | {q2} |




NFA without ∈

Transition diagram

Reference:

- Introduction to the Theory of Computation” by Michael Sipser.

## Related Posts:

1. Definition of Deterministic Finite Automata
2. Notations for DFA
3. How do a DFA Process Strings?
4. DFA solved examples
5. Definition Non Deterministic Finite Automata
6. Moore machine
7. Mealy Machine
8. Regular Expression Examples
9. Regular expression
10. Arden’s Law
11. NFA with ∈ to DFA Indirect Method
12. Define Mealy and Moore Machine
13. What is Trap state ?
14. Equivalent of DFA and NFA
15. Properties of transition functions
16. Mealy to Moore Machine
17. Moore to Mealy machine
18. Diiference between Mealy and Moore machine
19. Pushdown Automata
20. Remove ∈ transitions from NFA

21. TOC 1
22. Diiference between Mealy and Moore machine
23. RGPV TOC What do you understand by DFA how to represent it
24. What is Regular Expression
25. What is Regular Set in TOC
26. RGPV short note on automata
27. RGPV TOC properties of transition functions
28. RGPV TOC What is Trap state
29. DFA which accept 00 and 11 at the end of a string
30. CFL are not closed under intersection
31. NFA to DFA | RGPV TOC
32. Moore to Mealy | RGPV TOC PYQ
33. DFA accept even 0 and even 1 |RGPV TOC PYQ
34. Short note on automata | RGPV TOC PYQ
35. DFA ending with 00 start with 0 no epsilon | RGPV TOC PYQ
36. DFA ending with 101 | RGPV TOC PYQ
37. Construct DFA for a power n, n>=0 || RGPV TOC
38. Construct FA divisible by 3 | RGPV TOC PYQ
39. Construct DFA equivalent to NFA | RGPV TOC PYQ
40. RGPV Define Mealy and Moore Machine
41. RGPV TOC Short note on equivalent of DFA and NFA
42. RGPV notes Write short note on NDFA
43. Minimization of DFA
44. Construct NFA without ∈
45. CNF from S–>aAD;A->aB/bAB;B->b,D->d.
46. NDFA accepting two consecutive a’s or two consecutive b’s.
47. Regular expresion to CFG

48. Regular expression to Regular grammar
49. Grammar is ambiguous. $S \rightarrow aSbS|bSaS|\in$
50. leftmost and rightmost derivations
51. Construct Moore machine for Mealy machine
52. RGPV TOC PYQs
53. Introduction to Automata Theory
54. Design a NFA that accepts the language over the alphabet, $\Sigma = \{0, 1, 2\}$ where the decimal equivalent of the language is divisible by 3.