Al-Hamdaniya University

College of Education

Computer Science

Stage: 2nd

Conversion from NFA to DFA

An NFA can have zero, one or more than one move from a given state on a given input symbol. An NFA can also have NULL moves (moves without input symbol). On the other hand, DFA has one and only one move from a given state on a given input symbol.

Steps for converting NFA to DFA:

Step 1: Convert the given NFA to its equivalent transition table

To convert the NFA to its equivalent transition table, we need to list all the states, input symbols, and the transition rules. The transition rules are represented in the form of a matrix, where the rows represent the current state, the columns represent the input symbol, and the cells represent the next state.

Step 2: Create the DFA's start state

The DFA's start state is the set of all possible starting states in the NFA.

Step 3: Create the DFA's transition table

The DFA's transition table is similar to the NFA's transition table, but instead of individual states, the rows and columns represent sets of states.

Step 4: Create the DFA's final states

The DFA's final states are the sets of states that contain at least one final state from the NFA.

Example: Consider the following NFA shown in Figure 1.

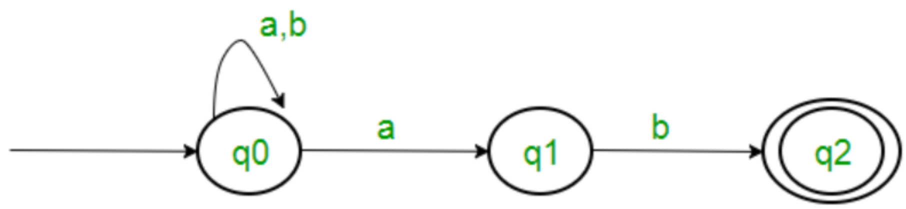


Figure 1

Following are the various parameters for NFA. Q = { q0, q1, q2 } in put (a, b)

State	a	b
q0	q0,q1	q0
q1		q2
q2		

Step 1: Q' = ?

Step 2: Q' = {q0}

Step 3: For each state in Q',

find the states for each input symbol.

Currently, state in Q' is q0, find moves from q0 on input symbol a and b using transition function of NFA and update the transition table of DFA.

State	a	b
q0	{q0,q1}	q0

Now { q0, q1 } will be considered as a single state.

As its entry is not in Q’, add it to Q’.

So Q’ = { q0, { q0, q1 } } Now, moves from state { q0, q1 } on different input symbols are not present in transition table of DFA,

we will calculate it like: ?’ ({ q0, q1 }, a) = ? (q0, a) ? ? (q1, a) = { q0, q1 } ?’ ({ q0, q1 }, b) = ? (q0, b) ? ? (q1, b) = { q0, q2 } Now we will update the transition table of DFA.

State	a	B
q0	{q0,q1}	q0
{q0,q1}	{q0,q1}	{q0,q2}

Now { q0, q2 } will be considered as a single state.

As its entry is not in Q’, add it to Q’.

So Q’ = { q0, { q0, q1 }, { q0, q2 } } Now, moves from state {q0, q2} on different input symbols are not present in transition table of DFA,

we will calculate it like: ?’ ({ q0, q2 }, a) = ? (q0, a) ? ? (q2, a) = { q0, q1 } ?’ ({ q0, q2 }, b) = ? (q0, b) ? ? (q2, b) = { q0 } Now we will update the transition table of DFA.

State	a	B
q0	{q0,q1}	q0
{q0,q1}	{q0,q1}	{q0,q2}
{q0,q2}	{q0,q1}	q0

As there is no new state generated, we are done with the conversion.

Final state of DFA will be state which has q2 as its component i.e., { q0, q2 } Following are the various parameters for DFA.

Q' = { q0, { q0, q1 }, { q0, q2 } } ? = (a, b) F = { { q0, q2 } } and transition function ?' as shown above. The final DFA for above NFA has been shown in Figure 2.

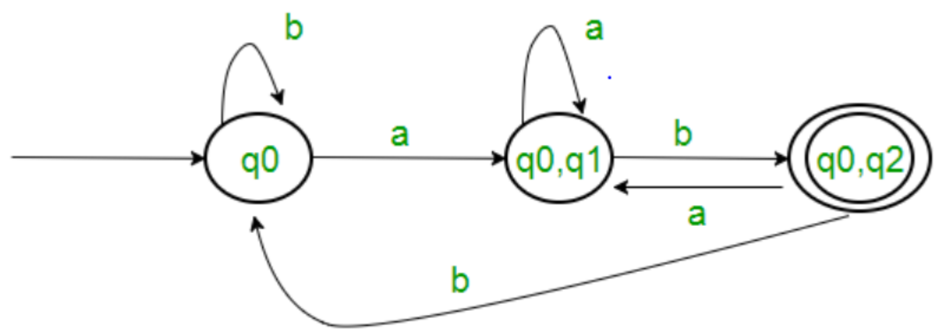
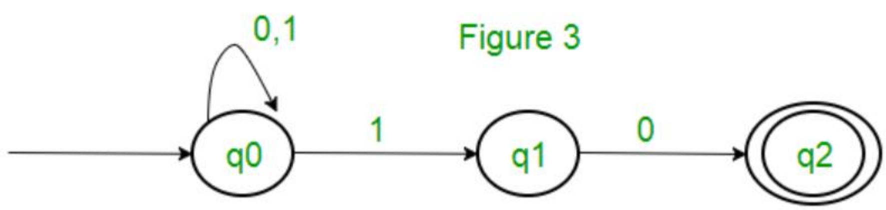


Figure 2

H.W: convert NFA to DFA

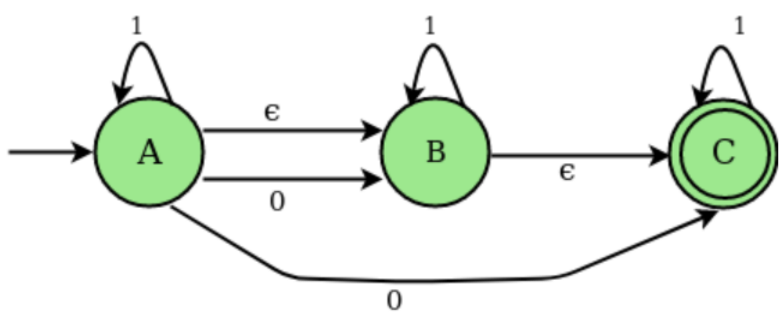
Figure 3



Convert NFA with epsilon move to DFA

Non-deterministic Finite Automata (NFA): NFA is a finite automation where for some cases when a single input is given to a single state, the machine goes to more than 1 states, i.e. some of the moves cannot be uniquely determined by the present state and the present input symbol.
NFA with (null) or ϵ move: If any finite automata contains ϵ move or transaction, then that finite automata is called NFA with ϵ moves.

Example : Consider the following figure of NFA with ? move :



Transition state table for the above NFA

STATES	0	1	epsilon
A	B, C	A	B
B	–	B	C
C	-	C	–

```
? closure(A) : {A, B, C}
? closure(B) : {B, C}
? closure(C) : {C}
```

Deterministic Finite Automata (DFA) : DFA is a finite automata where, for all cases, when a single input is given to a single state, the machine goes to a single state, i.e., all the moves of the machine can be uniquely determined by the present state and the present input symbol.

Step 1 : Take ? closure for the beginning state of NFA as beginning state of DFA.

Step 2 : Find the states that can be traversed from the present for each input symbol (union of transition value and their closures for each states of NFA present in current state of DFA).

Step 3 : If any new state is found take it as current state and repeat step 2.

Step 4 : Do repeat Step 2 and Step 3 until no new state present in DFA transition table.

Step 5 : Mark the states of DFA which contains final state of NFA as final states of DFA.

Transition State Table for DFA corresponding to above NFA

STATES	0	1
A, B, C	B, C	A, B, C
B, C	C	B, C
C		C

DFA STATE DIAGRAM