CS 208 : Automata Theory and Logic DATE

Lecture - 15

Topic: Representation of DFA and NFA

Scribed by: Yashwanth VVS (22B0970)
Checked and compiled by:

Disclaimer. Please note this document has not received the usual scrutiny that formal publications enjoy. This may be distributed outside this class only with the permission of the instructor.

DFA Representation

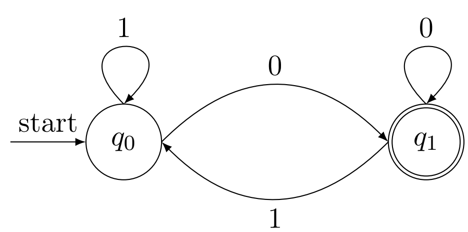
A Deterministic Finite Automaton(DFA) is represented as follows:

$$(Q, \Sigma, q_0, \delta, F)$$

where,

Q is the set of all possible states
Σ is the alphabet
$q_0(\in Q)$ is the initial state
$\delta(: Q \times \Sigma \to Q)$ is the transition function
$F(\subseteq Q)$ is the set of final states

Example(DFA)



This DFA can be represented as $(\{\mathbf{q_0}, \mathbf{q_1}\}, \{\mathbf{0}, \mathbf{1}\}, \mathbf{q_0}, \delta, \{\mathbf{q_1}\})$.
Here, δ, the transition function is defined as:

$\delta(q_0, 0) = q_1$
$\delta(q_0, 1) = q_0$
$\delta(q_1, 0) = q_1$
$\delta(q_1, 1) = q_0$

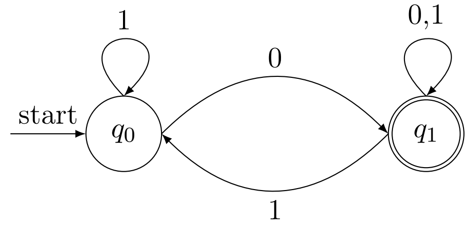
NFA Representation

A Non-deterministic Finite Automaton(NFA) has a very similar representation:

$$(Q, \Sigma, Q_0, \delta, F)$$

where $Q_0(\subseteq Q)$ is the set of initial states and $\delta(: Q \times \Sigma \to 2^Q)$ is the transition function.

Example(NFA)



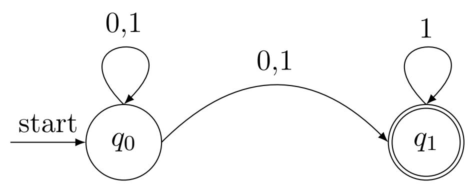
This NFA can be represented as $(\{\mathbf{q_0}, \mathbf{q_1}\}, \{\mathbf{0}, \mathbf{1}\}, \{\mathbf{q_0}\}, \delta, \{\mathbf{q_1}\})$.
Here, δ, the transition function is defined as:

$$\delta(q_0, 0) = \{q_1\}$$
$$\delta(q_0, 1) = \{q_0\}$$
$$\delta(q_1, 0) = \{q_1\}$$
$$\delta(q_1, 1) = \{q_0, q_1\}$$

In an NFA, the range of δ(transition function) is not Q but 2^Q (power set of Q). This is because for each combination of state and input symbol, the transition function can potentially map to multiple states belonging to Q, that is, map to a subset of Q.

Conversion of NFA to DFA

Now we'll see how to convert a NFA to a DFA through an example.
We call the following NFA 'A'



The language depicted by this NFA is all the strings formed using {0,1} excluding the empty string(ϵ).
We represent this as:

$$L(A) = \{w \in \{0, 1\}^* | w \text{ is accepted by A}\}$$

that is,

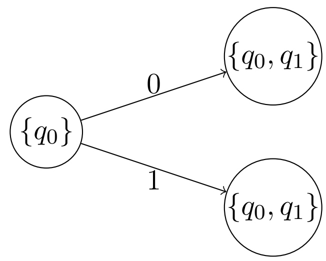
$$L(A) = \Sigma^* \backslash \{\epsilon\}$$

The transition function(δ) table looks as follows:

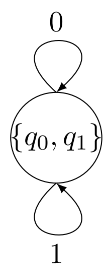
Q	Σ	2^Q
q_0	0	$\{q_0, q_1\}$
q_0	00	$\{q_0, q_1\}$
q_0	01	$\{q_0, q_1\}$

To convert this NFA to a DFA, we need to track the states that can be reached after n choices which will be a subset of 2^Q.

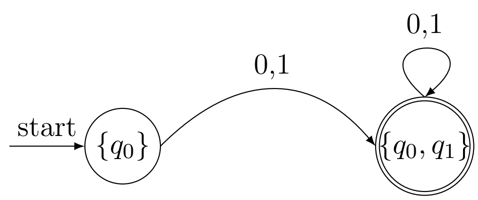
Step 1



Step 2



Final DFA



Call this DFA 'A''.
One property we've extensively used in the conversion is:

$$\textbf{If } S \subseteq Q\textbf{, then } \delta(S, 0) = \bigcup_{q \in S} \delta(q, 0)$$

Now, to show that the languages represented by the NFA(A) and the DFA(A') are the same, i.e, $L(A) = L(A')$, we need to show the following:

1. $L(A) \subseteq L(A')$
2. $L(A') \subseteq L(A)$