4CS4-6: Theory of Computation (Equivalence of NFA and DFA)

Lecture 05: Jan. 27 & 29, 2020

Prof. K.R. Chowdhary *: Professor of CS*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

5.1 Introduction

For a language L, if a NFA recognizes it then there also exists an equivalent DFA that recognizes it. In fact, in this DFA only certain sequences of transitions are to be selected for the given input to reach to final state, and other sequence of states though exists, are not accounted for final walk-through. Therefore, the class of languages accepted by a DFA is subset of the class of languages accepted by an NFA. It will be shown that these classes are in fact equal. An NFA though makes available a much more generality than DFA, but is in no way more powerful than the corresponding DFA. An NFA can always be converted into an equivalent DFA.

5.2 Equivalence of NFA and DFA

Any two finite automaton M_1 and M_2 are equivalent if the languages accepted by them are equivalent, i.e., $L(M_1) = L(M_2)$. The following theorem shows that for every NFA, there exists an equivalent DFA.

Theorem 5.1 *For every NFA there exists an equivalent DFA.*

Proof. Let $M_N = (Q, \Sigma, \delta, s, F)$ be an NFA, and we intend to show that there exists a DFA $M_D = (Q', \Sigma, \delta', s', F')$ equivalent to M_N. To prove this we need to represent tuples of M_D in terms of the tuples of M_N.

We know that after each symbol is read, an NFA occupies a set of states out of 2^Q. If each set is considered as a single state for an equivalent DFA, the maximum number of states in M_D will be $Q' = 2^Q$. Accordingly, we denote an element of Q' by $\{\dots, q_i, \dots\}$, where $q_i \in Q$.

The definition of transition function δ' in M_D is bit complex. A move in an equivalent M_D on reading an input symbol $a \in \Sigma$ simulates a transition in M_N associated with possibly number of *null* (ε) transitions in M_u. For any state $q_i \in Q$, we define $\mathcal{C}(q_i)$ as ε-*closure* of q_i as a set of all the states of M_N that are reachable from state q_i with ε input. Hence,

$$\mathcal{C}(q_i) = \{q_j \in Q \mid \delta(q_i, \varepsilon) = q_j\}. \tag{5.1}$$

In other words, $\mathcal{C}(q_i)$ is ε-*closure* of the set $\{q_i\}$ under the relation,

$$\begin{aligned}\{(q_j, q_k) \mid \ &\text{there is a transition of the form } \delta(q_j, \varepsilon) = q_k;\\ &q_j \in \mathcal{C}(q_i), \ \text{and } q_k \notin \mathcal{C}(q_i)\}.\end{aligned}$$

The set $\mathcal{C}(q_i)$ is computed using algorithm 5.1.

Algorithm 1 *ε-Closure.*

```
for $i$:= 1 to $n$ do
    $\mathcal{C}(q_i) := \{q_i\}$
end for
while there is transition $\delta(q_j, \varepsilon) = q_k; q_j \in \mathcal{C}(q_i)$ and $q_k \notin \mathcal{C}(q_i)$ do
    $\mathcal{C}(q_i) := \mathcal{C}(q_i) \cup \{q_k\}$
end while
```

This algorithm adds an edge to $\mathcal{C}(q_i)$ in every pass through the while loop. The transition graph thus formed for the DFA M_D can have at most $2^{|Q|} \times |\Sigma|$ number of edges, hence the loop should eventually terminate.

Based on above, different tuples of the equivalent deterministic finite automata M_D can now be defined as,

$$\begin{aligned}&Q' = 2^Q\\ &s' = \mathcal{C}(s), \ (\varepsilon\text{-closure of s})\\ &F' = \{D\}.\end{aligned}$$

In the above, every element D of F' is defined as, if $q_f \in F$, then $D = \{\dots, q_f, \dots\} \in F'$. In addition, for each $a \in \Sigma$ and for each $D \in F'$ the δ' can be generalized as follows:

$$\delta'(D, a) = \bigcup \{\{q_j\} \mid \delta(q_j, a) = q_k; q_j \in D \text{ and } q_k \in Q\} \tag{5.2}$$

where $\{q_j\} = \mathcal{C}(q_j)$, is set of all the states reachable from q_j through ε - transitions. Hence, the $\delta'(D, a)$ is a state of M_D corresponding to the set of all the states of M_N to which M_N can move by reading an input a followed by possibly any number of ε-transitions. The set of these Ds form the final states in equivalent DFA.

Since all the tuples for a DFA have been defined in terms of the tuples of NFA, it can be concluded that there exists an equivalent DFA consisting these tuples, for every NFA. ■

5.3 Examples

Following example demonstrates the conversion of NFA for a given DFA.

Example 5.2 *Given the NFA in figure 5.1 find an equivalent DFA.*

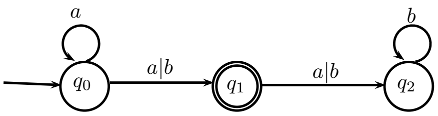

Figure 5.1: NFA.

Let the transition function for *NFA* be δ, for equivalent DFA as δ', and start state of equivalent DFA as $\{q_0\}$.

$$\delta'(\{q_0\}, a) = \{q_0, q_1\}$$
$$\delta'(\{q_0\}, b) = \{q_1\}.$$

Note that, in above we have used $\{q_0\}$, $\{q_2\}$, as sets. This is because in δ' there are state sets only, and each sets as subset of Q.

$$\begin{aligned}\delta'(\{q_0, q_1\}, a) &= \delta(q_0, a) \cup \delta(q_1, a)\\ &= \{q_0, q_1\} \cup \{q_2\}\\ &= \{q_0, q_1, q_2\}.\end{aligned}$$

$$\begin{aligned}\delta'(\{q_0, q_1\}, b) &= \delta(q_0, b) \cup \delta(q_1, b)\\ &= \{q_1\} \cup \{q_2\}\\ &= \{q_1, q_2\}.\end{aligned}$$

Proceeding in the similar way we compute the other states, and finally obtain the equivalent DFA shown in figure 5.2.

$$\delta'(\{q_1\}, a) = \{q_2\}$$
$$\delta'(\{q_1\}, b) = \{q_2\}.$$

$$\begin{aligned}\delta'(\{q_0, q_1, q_2\}, a) &= \delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a)\\ &= \{q_0, q_1\} \cup \{q_2\} \cup \phi\\ &= \{q_0, q_1, q_2\}.\end{aligned}$$

$$\begin{aligned}\delta'(\{q_0, q_1, q_2\}, b) &= \delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q2, b)\\ &= \{q_1\} \cup \{q_2\} \cup \{q_2\}\\ &= \{q_1, q_2\}.\end{aligned}$$

$$\begin{aligned}\delta'(\{q_1, q_2\}, a) &= \delta(q_1, a) \cup \delta(q_2, a)\\ &= \{q_2\} \cup \phi\\ &= \{q_2\}.\end{aligned}$$

$$\begin{aligned}\delta'(\{q_1, q_2\}, b) &= \delta(q_1, b) \cup \delta(q_2, b)\\ &= \{q2\} \cup \phi\\ &= \{q_2\}.\end{aligned}$$

$$\begin{aligned}\delta'(\{q_2\}, a) &= \delta(q_2, a) = \phi\\ \delta'(\{q_2\}, b) &= \delta(q_2, b) = \{q_2\}.\end{aligned}$$

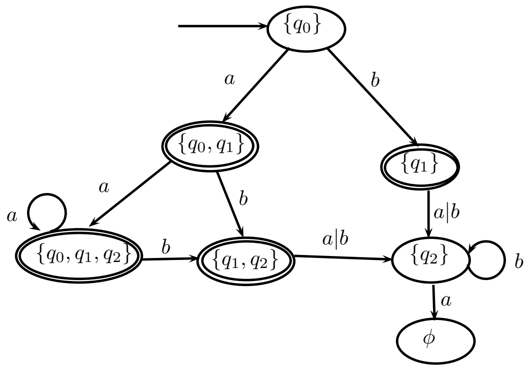

Figure 5.2: Equivalent DFA for the NFA in fig. 5.1.

We note that number of states in the equivalent DFA are 7. The maximum states in this case can be $2^{\{q_0,q_1,q_2\}} = 8$.

A state set label in the equivalence *DFA* is final state if one of the state q_i in this set is final state of corresponding *NFA*. For example, if $q_j \in F$ in NFA, then $\{\dots, q_j, \dots\} \in F'$ is in DFA. The empty states can be dropped as they do not have any practical significance.