4CS4-6: Theory of Computation (Closure properties of Regular Languages)

Lecture 07: Feb. 03 & 06, 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.*

7.1 Introduction

We know that class of languages accepted by finite automaton are only those represented by regular expressions, i.e., regular languages. Now our approach will be to show that there is an NFA with ε-transitions, which accepts these languages. We have learned in previous chapters that for every NFA there exists an equivalent DFA. We will also show that for every DFA there exists a regular expression that represents the language of that DFA. The above discussion leads to a cycle like phenomena shown in figure 7.1. The diagram indicates that corresponding to a regular expression there is a regular language, corresponding to a regular language there is an NFA, corresponding to an NFA there exist a DFA, and finally corresponding to a DFA there exists a regular expression. To prove the above representation of cyclic sequence, we need to show that every regular language corresponds to an NFA with ε-transitions. Since, every regular language corresponds to a regular expression, we arrive at a proof that for every regular expression there exists an NFA, which recognizes it.

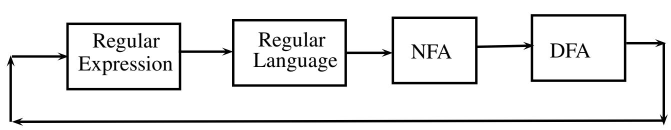

Figure 7.1: Equivalence of Regular Expression, Regular Language, NFA, and DFA.

7.2 Step-by-step Construction of FA

We discussed in the previous chapters that regular expressions are formed by *union*, *concatenation*, and *Kleene Star* operation on regular expressions. In other words a regular expression is closed under the operations of union, concatenation and Kleene Star. Hence, the *class of regular languages* should also be closed under the operations of union concatenation and Kleene Star. To prove this, an approach used here will be similar, i.e., applying the operations of union, concatenation, and Kleene Star on class of languages accepted by finite automation.

Theorem 7.1 *Prove that class of languages accepted by finite automata are closed under the operation of union, concatenation, and Kleene Star.*

Proof: For each of the three cases, given two automata M_1 and M_2

$$M_1 = (Q_1, \Sigma_1, \delta_1, s_1, F_1),$$
$$M_2 = (Q_2, \Sigma_2, \delta_2, s_2, F_2).$$

we construct an automata M, taking each case individually. The simplified diagrams of M_1, M_2 are shown in figure 7.2. For any input x, the automata M_1 starts with s_1, followed with any number of transitions, then ultimately leading to one of the final states in F_1. Thus, for $x \in L(M_1)$, there is $\delta_1^*(s_1, x) \in F_1$. In the similar way, the automata M_2 accepts.

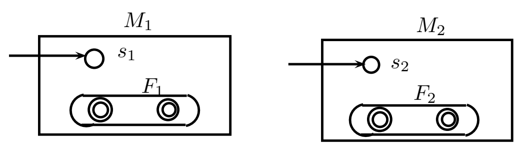

Figure 7.2: Automata M_1 and M_2.

(a) Union of two regular languages: Let M_1 and M_2 are DFAs accepting the languages $L(M_1)$ and $L(M_2)$, respectively. Therefore, the language accepted by resultant automata M will be $L(M) = L(M_1) \cup L(M_2)$.

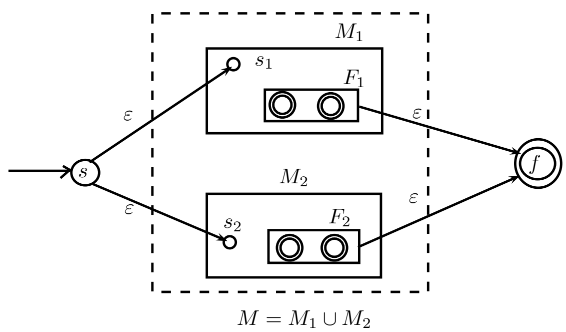

Figure 7.3: Automata $M = M_1 \cup M_2$.

Figure 7.3 shows the automaton $M = (Q, \Sigma, \delta, s, F)$ constructed using automata M_1 and M_2. For an input string w, every path of sequence of states in M will start from s, and then there will be a ε-transition to either to s_1 or s_2. If there is input $x \in L(M_1)$ then there is a transition to s_1, and then rest of the states are traversed in M_1, followed with a transition to F_1, followed with a ε-transition to f. Otherwise, if input is $y \in L(M_2)$ then, there is a ε-transition to s_2, then the traversing of path would be through M_2 until the final state(s) F_2 of M_2 is reached. This is followed with a ε-transition to f. Hence, M accepts the string $\{w\} = \{x\} \cup \{y\}$, if, M_1 or M_2 accepts it. Therefore, $L(M) = L(M_1) \cup L(M_2)$.

Let input in the machine M be at state s, and from here it makes a guess to choose a ε-transition to s_1 or s_2. The components of M can be defined in terms of the components of M_1 and M_2 as follows:

$$Q = Q_1 \cup Q_2 \cup \{s, f\}$$
$$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{\varepsilon\}$$
δ for any $q \in Q$ and $a \in \Sigma$ is:

$$\delta(q,a) = \begin{cases} \delta_1(q,a), \text{ if } q \in Q_1 \text{ and } a \in \Sigma_1, \\ \delta_2(q,a), \text{ if } q \in Q_2 \text{ and } a \in \Sigma_2, \\ \delta(q,a) = \{s_1\} \cup \{s_2\}, \text{ if } q = s \text{ and } a = \varepsilon \\ f, \text{ if } q \in F_1 \cup F_2 \text{ and } a = \varepsilon \end{cases}$$

and $F = \{f\}$.

Considering that $x \in L(M_1)$ and $y \in L(M_2)$, the language $L(M)$ of finite automaton M can be defined as:

$$L(M) = \{\{x\} \cup \{y\} \mid x \in L(M_1) \wedge y \in L(M_2)\}. \tag{7.1}$$

If input is $x \in L(M_1)$ at s, it will be nondeterministically decided to process x in M_1. Alternatively, for input $y \in L(M_2)$, the processing will be through M_2.

(b) Concatenation of two regular languages: In this case again we consider M_1, M_2 as two nondeterministic finite automata, shown in figure 7.2, and construct a combination NFA M shown in figure 7.4, such that $L(M) = L(M_1) \circ L(M_2)$.

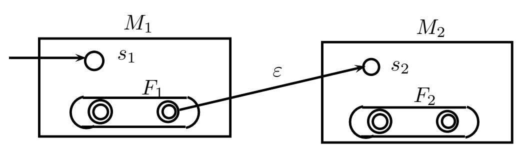

Figure 7.4: Finite Automata $M = M_1 \circ M_2$.

Let the finite automata M_1, M_2 are defined as earlier, and we define the combination automaton M in terms of M_1 and M_2 as follows:

Let $M = (Q, \Sigma, \delta, s, F)$.

$$Q = Q_1 \cup Q_2$$
$$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{\varepsilon\}$$

The δ for $q \in Q$ and $a \in \Sigma$ is:

$$\delta(q,a) = \begin{cases} \delta_1(q,a), \text{ if } q \in Q_1 \text{ and } a \in \Sigma_1, \\ \delta_2(q,a), \text{ if } q \in Q_2 \text{ and } a \in \Sigma_2, \\ \delta(q,a) = s_2, \text{ if } q \in F_1 \text{ and } a = \varepsilon. \end{cases}$$

$s = s_1$, and, $F = F_2$.

For any string $w = xy$ applied as input to M, where $x \in L(M_1)$ and $y \in L(M_2)$, there will be a path from s_1 to f_1 labeled by string x, then there will be a ε-transition from f_1 to s_2, then there will be a path from s_2 to f_2 labeled by y. Therefore, $L(M) = \{xy \mid x \in L(M_1) \text{ and } y \in L(M_2)\}$. Hence, $L(M) = L(M_1) \in L(M_2)$.

(c) Kleene Star on $L(M_1)$: Let M_1 be a nondeterministic finite automaton. Using this we want to construct an NFA M such that $L(M) = (L(M_1))^*$. For example, for $^* = 3$, there is $L(M) = L(M_1) \circ L(M_1) \circ L(M_1)$. Therefore, Kleene Star is a concatenation operation of language of a FA itself, with that language, any number of times. The resultant NFA M obtained after * operation on M_1 is shown in figure 7.5.

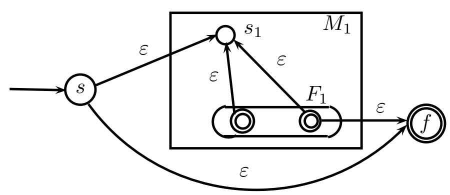

Figure 7.5: Finite Automata $M = M_1^*$ constructed using Kleene Star operation on M_1.

It should be noted that a separate initial state s has been added in M to cause a ε-transition from s to s_1 corresponds to Kleene star operation $(L)^*$. The Kleene star has the effect of ε-transitions from all the final states F in M to start state s_1 of M_1.

If the finite automata M_1 is represented by $M_1 = (Q_1, \Sigma_1, \delta_1, s_1, F_1)$ then the finite automata $M = (Q, \Sigma, \delta, s, F)$ can be represented in terms of M_1 as:

$$Q = Q_1 \cup \{s\},$$
$$\Sigma = \Sigma_1 \cup \{\varepsilon\}.$$

The transition function δ for M is defined as:

$$\delta(q, a) = \begin{cases} \delta_1(q, a), \text{ if } q \in Q_1 \text{ and } a \in \Sigma_1, \\ s_2, \text{ if } q \in F_1 \text{ and } a = \varepsilon, \\ \{s_1, f\}, \text{ if } q = s \text{ and } a = \varepsilon. \end{cases}$$

and $F = \{f\}$.

Therefore, $L(M) = L(M_1)^*$. This concludes the proof that a class of languages accepted by finite automata are closed under the operation of *union*, *concatenation* and *Kleene Star*.