Unit-II

After going through this chapter, you should be able to understand :

- Regular sets and Regular Expressions
- Identity Rules
- Constructing FA for a given REs
- Conversion of FA to REs
- Pumping Lemma of Regular sets
- Closure properties of Regular sets

3.1 REGULAR SETS

A special class of sets of words over S, called regular sets, is defined recursively as follows. (Kleene proves that any set recognized by an FSM is regular. Conversely, every regular set can be recognized by some FSM.)

1. Every finite set of words over S (including $\in$, the empty set) is a regular set.
2. If A and B are regular sets over S, then $A \cup B$ and AB are also regular.
3. If S is a regular set over S, then so is its closure S*.
4. No set is regular unless it is obtained by a finite number of applications of definitions (1) to (3).

i.e., the class of regular sets over S is the smallest class containing all finite sets of words over S and closed under union, concatenation and star operation.

Examples:

i) Let $\Sigma = \{a,b\}$ then the set of strings that contain both odd number of a's and b's is a regular set.

ii) Let $\Sigma = \{0\}$ then the set of strings $\{0,00,000,.....\}$ is a regular set.

iii) Let $\Sigma = \{0,1\}$ then the set of strings $\{01,10\}$ is a regular set.

3.2 REGULAR EXPRESSIONS

The languages accepted by FA are regular languages and these languages are easily described by simple expressions called regular expressions. We have some algebraic notations to represent the regular expressions.

Regular expressions are means to represent certain sets of strings in some algebraic manner and regular expressions describe the language accepted by FA.

If Σ is an alphabet then regular expression(s) over this can be described by following rules.

1. Any symbol from $\Sigma, \in$ *and* ϕ are regular expressions.
2. If r_1 and r_2 are two regular expressions then *union* of these represented as $r_1 \cup r_2$ or $r_1 + r_2$ is also a regular expression
3. If r_1 and r_2 are two regular expressions then *concatenation* of these represented as r_1r_2 is also a regular expression.
4. The Kleene closure of a regular expression r is denoted by $r*$ is also a regular expression.
5. If r is a regular expression then (r) is also a regular expression.
6. The regular expressions obtained by applying rules 1 to 5 once or more than once are also regular expressions.

Examples :

(1) If $\Sigma = \{a, b\}$, then

(a) a is a regular expression (Using rule 1)
(b) b is a regular expression (Using rule 1)
(c) $a + b$ is a regular expression (Using rule 2)
(d) $b*$ is a regular expression (Using rule 4)
(e) ab is a regular expression (Using rule 3)
(f) $ab + b*$ is a regular expression (Using rule 6)

(2) Find regular expression for the following

(a) A language consists of all the words over $\{a, b\}$ ending in b.
(b) A language consists of all the words over $\{a, b\}$ ending in bb.
(c) A language consists of all the words over $\{a, b\}$ starting with a and ending in b.
(d) A language consists of all the words over $\{a, b\}$ having bb as a substring.
(e) A language consists of all the words over $\{a, b\}$ ending in aab.

Solution : Let $\Sigma = \{a,b\}$, and

All the words over $\Sigma = \{\in, a, b, aa, bb, ab, ba, aaa, \ldots\} = \Sigma * \; or \; (a + b) * \; or \; (a \cup b) *$

$= (\{\in, a, b, aa, bb, \dots\})^*$

$= \{\in, a, b, aa, bb, ab, ba, aaa, bbb, abb, baa, aabb, \dots\}$

$= \{\text{All the words over } \{a, b\}\}$

$\equiv (a + b)^*$

So, $(a^* + b^*)^* \equiv (a + b)^*$

3.3 IDENTITIES FOR REs

The two regular expressions P and Q are equivalent (denoted as P = Q) if and only if P represents the same set of strings as Q does. For showing this equivalence of regular expressions we need to show some identities of regular expressions.

Let P, Q and R are regular expressions then the identity rules are as given below

1. $\in R = R \in = R$
2. $\in^* = \in$ $\in$ is null string
3. $(\phi)^* = \in$ ϕ is empty string.
4. $\phi R = R\phi = \phi$
5. $\phi + = R = R$
6. $R + R = R$
7. $RR^* = R^* R = R^+$
8. $(R^*)^* = R^*$
9. $\in + RR^* = R^*$
10. $(P+Q)R = PR + QR$
11. $(P+Q)^* = (P^*Q^*) = (P^* + Q^*)^*$
12. $R^*(\in + R) = (\in + R)R^* = R^*$
13. $(R + \in)^* = R^*$
14. $\in + R^* = R^*$
15. $(PQ)^* P = P(QP)^*$
16. $R^* R + R = R^* R$

3.3.1 Equivalence of two REs

Let us see one important theorem named Arden's Theorem which helps in checking the equivalence of two regular expressions.

Arden's Theorem : Let P and Q be the two regular expressions over the input set Σ. The regular expression R is given as

$$R = Q + RP$$

Which has a unique solution as $R = QP^*$

Proof : Let, P and Q are two regular expressions over the input string Σ.
If P does not contain $\in$ then there exists R such that

$$R = Q + RP \qquad \text{.... (1)}$$

We will replace R by QP* in equation 1.
Consider R. H. S. of equation 1.

$$= Q + QP^*P$$

$$= Q(\in + P^*P)$$

$$= QP^* \qquad \because \in + R^*R = R^*$$

Thus $\qquad R = QP^*$

is proved. To prove that $R = QP^*$ is a unique solution, we will now replace L.H.S. of equation 1 by Q + RP. Then it becomes

$$Q + RP$$

But again R can be replaced by Q + RP.

$$\therefore \qquad Q + RP = Q + (Q + RP)P$$

$$= Q + QP + RP^2$$

Again replace R by Q + RP.

$$= Q + QP + (Q + RP)P^2$$

$$= Q + QP + QP^2 + RP^3$$

Thus if we go on replacing R by Q + RP then we get,

$$Q + RP = Q + QP + QP^2 + \ldots\ldots + QP^i + RP^{i+1}$$

$$= Q(\in + P + P^2 + \ldots\ldots P^i) + RP^{i+1}$$

From equation 1,

$$R = Q(\in + P + P^2 + \ldots\ldots + P^i) + RP^{i+1} \qquad \text{... (2)}$$

Where $\qquad i \geq 0$

Consider equation 2,

$$R = Q\underbrace{(\in + P + P^2 + \ldots\ldots + P^i)}_{P^*} + RP^{i+1}$$

$$\therefore \qquad R = QP^* + RP^{i+1}$$

Let w be a string of length i.

$= \{\in, 0, 00, 1, 11, 111, 01, 10, \ldots\ldots\ldots\}$

$= \{\in$, any combination of 0's, any combination of 1's, any combination of 0 and 1 $\}$

Hence, L. H. S. = R. H. S. is proved.

3.4 RELATIONSHIP BETWEEN FA AND RE

There is a close relationship between a finite automata and the regular expression we can show this relation in below figure.

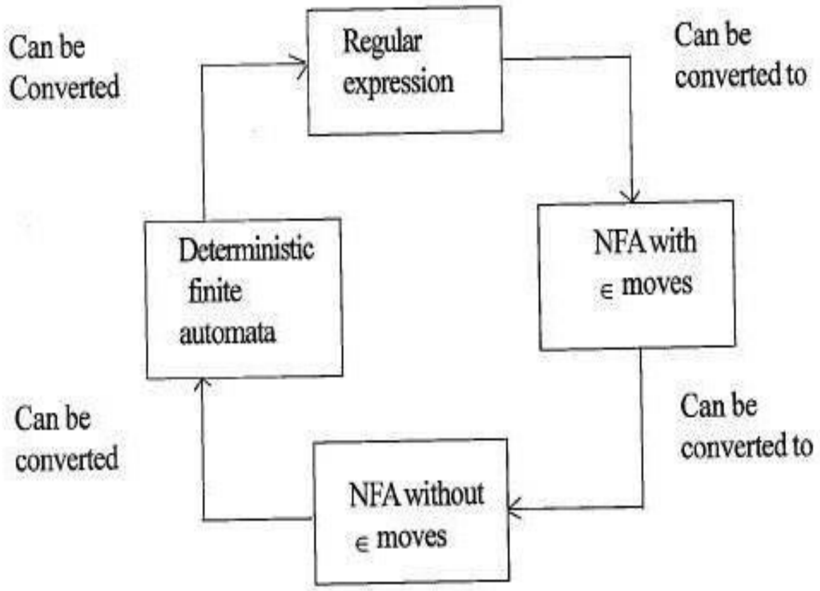


FIGURE : Relationship between FA and regular expression

The above figure shows that it is convenient to convert the regular expression to NFA with $\in$ moves. Let us see the theorem based on this conversion.

3.5 CONSTRUCTING FA FOR A GIVEN REs

Theorem : If r be a regular expression then there exists a NFA with $\in$ - moves, which accepts $L(r)$.

Proof : First we will discuss the construction of NFA M with $\in$ - moves for regular expression r and then we prove that $L(M) = L(r)$.

Let r be the regular expression over the alphabet Σ.

Construction of NFA with $\in$ - moves

Case 1 :

(i) $r = \phi$

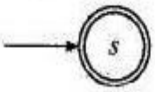
NFA $M = (\{s, f\}, \{\ \}, \delta, s, \{f\})$ as shown in Figure1 (a)

(No path from initial state s to reach the final state f.)

Figure 1 (a)

(ii) $r = \in$

NFA $M = (\{s\}, \{\ \}, \delta, s, \{s\})$ as shown in Figure 1 (b)

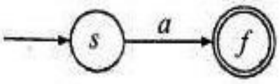


(The initial state s is the final state)

Figure 1 (b)

(iii) $r = a$, for all $a \in \Sigma$,

NFA $M = (\{s, f\}, \Sigma, \delta, s, \{f\})$



(One path is there from initial state s to reach the final state f with label a.)

Figure 1 (c)

Case 2 : $|r| \geq 1$

Let r_1 and r_2 be the two regular expressions over Σ_1, Σ_2 and N_1 and N_2 are two NFA for r_1 and r_2 respectively as shown in Figure 2 (a).

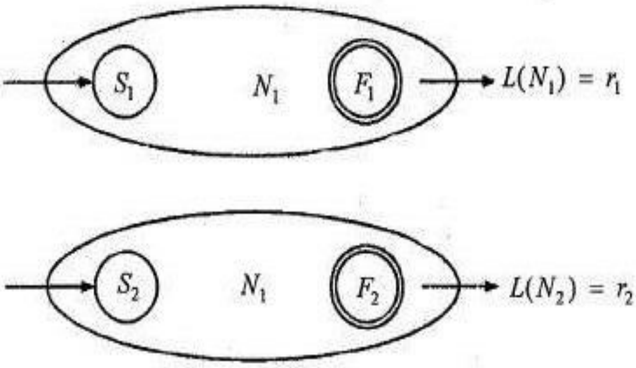


Figure 2 (a) NFA for regular expression r_1 and r_2

Now let us compute for final state, which denotes the regular expression.

r_{12}^2 will be computed, because there are total 2 states and final state is q_1 whose start state is q_0.

$$r_{12}^2 = (r_{12}^1)(r_{22}^1)^*(r_{22}^1) + (r_{12}^1)$$
$$= 0(\in)^*(\in) + 0$$
$$= 0 + 0$$

$r_{12}^2 = 0$ which is a final regular expression.

3.6.1 Arden's Method for Converting DFA to RE

As we have seen the Arden's theorem is useful for checking the equivalence of two regular expressions, we will also see its use in conversion of DFA to RE.

Following algorithm is used to build the r. e. from given DFA.

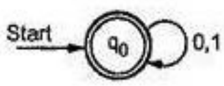
1. Let q_0 be the initial state.
2. There are $q_1, q_2, q_3, q_4, \ldots q_n$ number of states. The final state may be some q_j where $j \leq n$.
3. Let α_{ji} represents the transition from q_j to q_i.
4. Calculate q_i such that

   $$q_i = \alpha_{ji} . q_j$$

   If q_i is a start state

   $$q_i = \alpha_{ji} . q_j + \in$$
5. Similarly compute the final state which ultimately gives the regular expression r.

Example 1 : Construct RE for the given DFA.

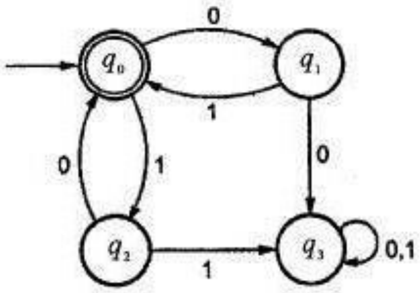


Solution :

Since there is only one state in the finite automata let us solve for q_0 only.

$$q_0 = q_0 0 + q_0 1 + \in$$
$$q_0 = q_0(0+1) + \in$$

Example 3 : Construct RE for the DFA given in below figure.



Solution : Let us see the equations

$$q_0 = q_1 1 + q_2 0 + \in$$
$$q_1 = q_0 0$$
$$q_2 = q_0 1$$
$$q_3 = q_1 0 + q_2 1 + q_3 (0+1)$$

Let us solve q_0 first,

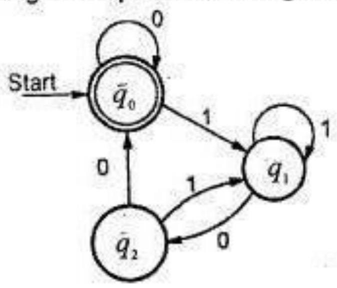
$$q_0 = q_1 1 + q_2 0 + \in$$
$$q_0 = q_0 01 + q_0 10 + \in$$
$$q_0 = q_0 (01+10) + \in \qquad \because R = Q + RP$$
$$q_0 = \in .(01+10)^* \qquad \Rightarrow QP^* \text{ where}$$
$$q_0 = (01+10)^* \qquad R = q_0, Q = \in, P = (01+10)$$

Thus the regular expression will be

$$r = (01+10)^*$$

Since q_0 is a final state, we are interested in q_0 only.

Example 4 : Find out the regular expression from given DFA.



Example 8 : Show that the language $L = \{a^i\ b^{2i} | i > 0\}$ is not regular.

Solution : The set of strings accepted by language L is,

$L = \{abb, aabbbb, aaabbbbbb, aaaabbbbbbbb...\}$

Applying Pumping lemma for any of the strings above.

Take the string abb.

It is of the form uvw.

Where, $|uv| \leq i, |v| \geq 1$

To find i such that $uv^i w \notin L$

Take i = 2 here, then

$uv^2 w = a(bb)b$

$= abbb$

Hence $uv^2 w = abbb \notin L$

Since abbb is not present in the strings of L.

$\therefore$ L is not regular.

Example 9 : Show that $L = \{0^n | n$ is a perfect square $\}$ is not regular.

Solution :

Step 1 : Let L is regular by Pumping lemma. Let n be number of states of FA accepting L.

Step 2 : Let $z = 0^n$ then $|z| = n \geq 2$.

Therefore, we can write z = uvw ; Where $|uv| \leq n, |v| \geq 1$.

Take any string of the language L = { 00, 0000, 000000 .... }

Take 0000 as string, here u = 0, v = 0, w = 00 to find i such that $uv^i w \notin L$.

Take i = 2 here, then

$uv^i w = 0(0)^2 00$

$= 00000$

This string 00000 is not present in strings of language L. So $uv^i w \notin L$.

$\therefore$ It is a contradiction.

3.9 PROPERTIES OF REGULAR SETS

Regular sets are closed under following properties.

1. Union
2. Concatenation

3. Kleene Closure
4. Complementation
5. Transpose
6. Intersection

1. Union : If R_1 and R_2 are two regular sets, then union of these denoted by $R_1 + R_2$ or $R_1 \cup R_2$ is also a regular set.

Proof : Let R_1 and R_2 be recognized by NFA N_1 and N_2 respectively as shown in Figure1(a) and Figure1(b).

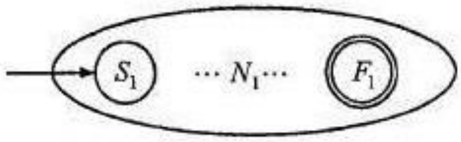


FIGURE 1(a) NFA for regular set R_1

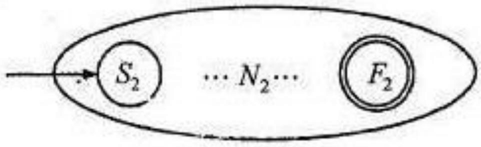


FIGURE 1(b) NFA for regular set R_2

We construct a new NFA N based on union of N_1 and N_2 as shown in Figure 1 (c)

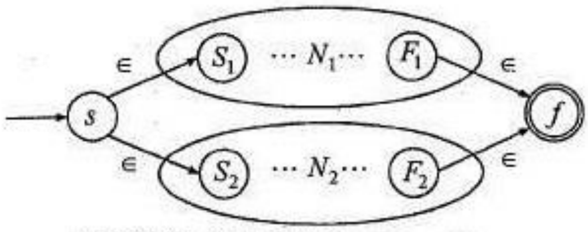


FIGURE 1(c) NFA for $N_1 + N_2$

Now,

$$L(N) = \in L(N_1) \in + \in L(N_2) \in$$
$$= \in R_1 \in + \in R_2 \in$$
$$= R_1 + R_2$$

Since, N is FA, hence $L(N)$ is a regular set (language). Therefore, $R_1 + R_2$ is a regular set.

2. **Concatenation :** If R_1 and R_2 are two regular sets, then concatenation of these denoted by R_1R_2 is also a regular set.

Proof : Let R_1 and R_2 be recognized by NFA N_1 and N_2 respectively as shown in Figure 2(a) and Figure 2(b).

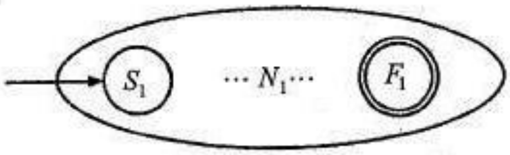


FIGURE 2(a) NFA for regular set R_1

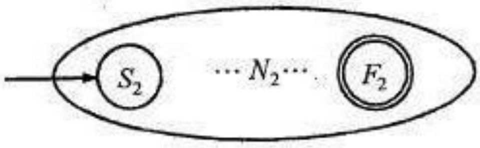


FIGURE 2(b) NFA for regular set R_2

We construct a new NFA N based on concatenation of N_1 and N_2 as shown in Figure2(c).

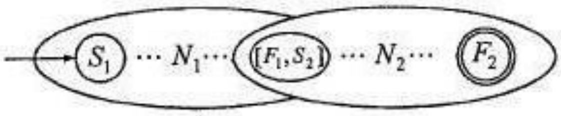


FIGURE 2(c) NFA for regular set R_1R_2

Now,

$L(N)$ = Regular set accepted by N_1 followed by regular set accepted by $N_2 = R_1R_2$

Since, $L(N)$ is a regular set, hence R_1R_2 is also a regular set.

3. **Kleene Closure :** If R is a regular set, then Kleene closure of this denoted by R^* is also a regular set.

Proof : Let R is accepted by NFA N shown in Figure 3(a).

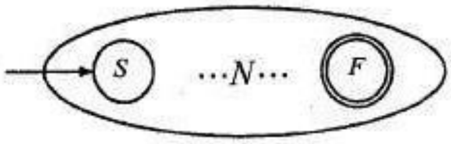


FIGURE 3(a) NFA for regular set R

We construct a new NFA based on NFA N as shown in Figure 3(b).

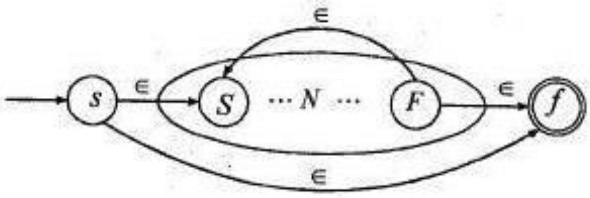


FIGURE 3(b) NFA for regular expression for R^*

Now,

$$L(N) = \{\in, R, RR, RRR, ...\}$$
$$= L^*$$

Since, $L(N)$ is a regular set, therefore R^* is a regular set.

4. Complement : If R is a regular set on some alphabet Σ, then complement of R is denoted by $\Sigma^* - R$ or $\overline{R}$ is also a regular set.

Proof : Let R be accepted by NFA $N = (Q, \Sigma, \delta, s, F)$. It means, $L(N) = R$. N is shown in Figure 4(a).

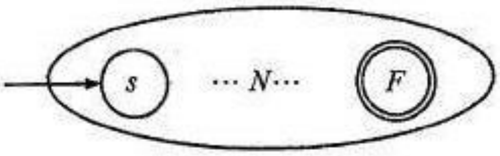


FIGURE 4(a) NFA for regular set R

We construct a new NFA N' based on N as follows :

(a) Change all final states to non-final states.

(b) Change all non-final states to final states.

N' is shown in Figure 4(b)

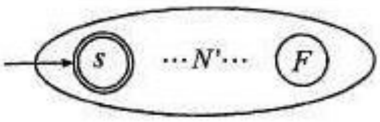


FIGURE 4 (b) NFA

Now,

$L(N') =$ {All the words which are not accepted by NFA N}

= { All the rejected words by NFA N}

$= \Sigma^* - R$

Since, $L(N')$ is a regular set, therefore $(\Sigma^* - R)$ is a regular set.

5. **Transpose :** If R is a regular set, then the transpose denoted by R^T, is also a regular set.

Proof : Let R be accepted by NFA $N = (Q, \Sigma, \delta, s, F)$ as shown in Figure 5(a).

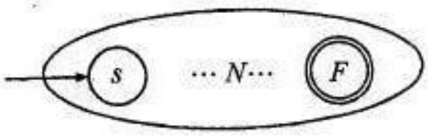


FIGURE 5 (a) NFA *N* for regular set *R*

If w is a word in R, then transpose (reverse) is denoted by w^T.

Let $w = a_1 a_2 ... a_n$

Then $w^T = a_n a_{n-1} .... a_1$

We construct a new N' based on N using following rules :

(a) Change the all final states into non-final states and merge all these into one state and make it initial state.

(b) Change initial state to final state.

(c) Reverse the direction of all edges.

N' is shown in Figure5 (b)

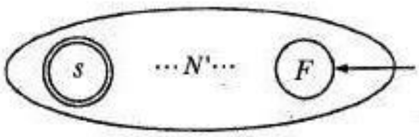


FIGURE 5(b) NFA *N'* for regular set R^T

Let $w = a_1 a_2 \ldots a_n$ be a word in R, then it is recognized by N and

$w^T = a_n a_{n-1} \ldots a_1$ is recognized by N' as shown in Figure5 (b)

In general, we say that if a word w in R is accepted by N, and then N' accepts w^T.

Since, $L(N')$ is a regular set containing all w^T; it means, $L(N') = R^T$.

Thus, R^T is a regular set.

6. Intersection : if R_1 and R_2 are two regular sets over Σ, then intersection of these denoted by $R_1 \cap R_2$ is also a regular set.

Proof : By De Morgan's law for two sets A and B over R,

$A \cap B = R^* - ((R^* - A) \cup (R^* - B))$

So, $R_1 \cap R_2 = \Sigma^* - ((\Sigma^* - R_1) \cup (\Sigma^* - R_2))$

Let $R_3 = (\Sigma^* - R_1)$ and $R_4 = (\Sigma^* - R_2)$

So, R_3 and R_4 are regular sets as these are complement of R_1 and R_2.

Let $R_5 = R_3 \cup R_4$

So, R_5 is a regular set because it is the union of two regular sets R_3 and R_4.

Let $R_6 = \Sigma^* - R_5$

So, R_6 is a regular set because it is the complement of regular set R_5.

Therefore, intersection of two regular sets is also regular set.

UNIT-3