**Problem 57**

**B** Yes, for any x, x L if and only if x contains one of the substrings aa, bb, abab, or baba. If x

contains neither aa nor bb, then x must consist of alternating b’s and a’s; therefore, if x

contains a non null substring ww, it must contain either abab or baba.

**D** No. We can prove this by using the pumping lemma, starting with a string of the form bnabn.

**G** No, because the set of palindromes cannot accepted by an FA.

**H** No. If L could be accepted by an FA, then {an | n ≥ 0}.

**Chapter 3:**

Problem 1

A aab or abb

B abab or baba

C bba

D abba

**Problem 2**

A aa

B ba

C a

D The shortest one is aba

**Problem 3**

**A**

Consider the regular expression r(r\*r + r\*) + r\*

Reducing the regular expression into its simpler form:

r(r\*r+r\*) + r\*

* rr\*r + rr\* + r\*
* r+r + r+ + r + r\* (Using the law, MM\* = M\*M = M+)
* r+ (r+ ∆) + r\* (Using identity law, M + ø =M)
* r+r + r\*
* rr\*r + r\* (Rewrite r+ = rr\*)
* r\*

The expression r\* represents either a single r or number of r’s. The first part of the expression consists of r\* and the second part of the expression also consists of r\*. So, the expression r(r\*r+r\*) + r\* can be represented as r\*.

Thus, the simpler regular expression for r(r\*r + r\*) + r\* is r\*.

**B**

Consider the regular expression (r + ∆)\*

Reducing the regular expression into its simpler form:

(r + ∆)\*

* (r)\* (Using identity law, M + ø =M)
* r\*

Thus, the simpler regular expression for (r + ∆)\* is r\*.

**C**

Consider the regular expression (r + s)\* rs(r + s)\* + s\*r\*

Reducing the regular expression into its simpler form:

(r + s)\* rs(r + s)\* + s\*r\*

* (r+s)\*rs(r+s)\*+(r+s) (Using the law, s\*r\* = r + s)
* (r+s)\*

The expression (r + s)\* represents either r, s or rs. So, the expression (r + s)\* rs(r + s)\* + s\*r\* can be represented as (r+s)\*.

Thus, the simpler regular expression for (r + s)\* rs(r + s)\* + s\*r\* is (r+s)\*.

**Problem 4**

Consider the following information:

The regular expression is (aa + aaa)(aa + aaa)\*.

The set of alphabets in the expression is ∑ = {a}.

For every integer n ≥ 2, there are non-negative integers I and j such that n = 2i + 3j.

Assume that r = (aa + aaa).

For i = 1 and j =1,

n = 2(1) + (1) over ∑

n = (aa + aaa), which is equal to r.

The value of n is 5. It is greater than or equal to 2. So, the condition n ≥ 2 is also satisfied.

Now, simplify the expression (aa + aaa)(aa + aaa)\* as follows:

(aa + aaa)(aa + aaa)\*

* nn\*
* n+ (Using the law, n+ = nn\*)

Thus, the simplied expression of (aa + aaa)(aa + aaa)\* is n+.

**Problem 7**

C) Λ + b + (a + bya + (a + b)\*bb

I) (a+b)\*(bb(a+b)\*aba+aba{a+b}\*bb){a+b}\*

J) (Λ+a+aa)(b+ba+baa)\*(1) (a+b)\*(bab(a+b)\*aba + aba(a+b)\*bab+baba+abab){a+b}\*

M)

**Problem 9**

Consider the statement “Every finite language is regular.”

Proof:

If the language is empty, the regular expression is defined as

If the language is finite and consists of the strings s1, s2, s3…sn for some positive integer ‘n’.

The regular expression for the finite language can be defined using any one of the following properties:

1. Union (U): The Union of strings can be represented as s1Us2Us3……Usn.
2. Concatenation(•): The concatenation of strings can be represented as s1•s2•s3……•sn.
3. Kleene star (\*): The Kleene star representation of strings can be represented as (s1)\*.
4. Kleene plus (+): The Kleene plus representation of strings can be represented as (s1)+.

Therefore, the finite language is regular language.