Mathematical Foundation for Computer Science

Prepared by: Er. Ankit Kharel

Nepal college of information technology

FINITE STATE AUTOMATA

- *Sequential Circuits and Finite state Machine*
- *Finite State Automata*
- *Non-deterministic Finite State Automata*
- *Language and Grammars*
- *Language and Automata*
- *Regular Expression*

GENERATING GRAMMAR FOR THE LANGUAGE:

1. **First write the Regular Expression for the language.**

Some Basic Rules:

a^* $A \rightarrow aA/\in$	$(a+b)^*$ $A \rightarrow aA/bA/\in$	$(a+b)^+$ $A \rightarrow aA/bA/a/b$

Q. Write the Grammar that generates string having Following Properties:

a) String of exactly length two

R.E. = (a+b)(a+b)

S→AA

A→a/b

b) String of at most length 2

R.E. = (a+b+∈)(a+b+∈)

S→AA

A→a/b/∈

c) Starts with a

R.E. = a $(a+b)^*$

S→aA

A→aA/bA/∈

c) Ends with ba

R.E. = $(a+b)^*$ba

S→Aba

A→aA/bA/∈

d) Stats with a and ends with b

R.E. = $a(a+b)^*b$

S→aAb

A→aA/bA/ ∈

e) Starts and ends with same symbol

R.E. = $a(a+b)^*a + b(a+b)^*b$

S→aAa/bAb/a/b

A→aA/bA/∈

f) Starts and ends with different symbol

R.E. = $a(a+b)^*b + b(a+b)^*a$

S→aAb/bAa

A→aA/bA/∈

c) Ends with ba

R.E. = $(a+b)^*$ba

S→Aba

A→aA/bA/∈

Q. Write the Context Free Grammar that generates Palindrome strong over $\sum(a, b)$

S→∈/a/b

S→aSa/bSb

Properties of regular language;

1. If L and M are regular languages, then L ∪ M(**UNION**) is a regular language.
   Let L and M be the languages of regular expressions R and S, respectively. Then R+S is a regular expression whose language is L U M.

2. If L is regular languages, then L^* **(Kleen Closure)** is a regular language.
   Let L the languages of regular expressions R. Then R^* is a regular expression whose language is L^*

3. If L and M are regular languages, then L.M**(Concatenation)** is a regular language.
   Let L and M be the languages of regular expressions R and S, respectively. Then R.S is a regular expression whose language is L.M.

4. If L is a regular language over Σ, then $\bar{L}$ **(Complement)** is also a regular language.
   Construct a DFA for L. This can be transformed into a DFA for $\bar{L}$ by making all accepting states non-accepting and vice versa.

5. If L and M are regular languages, then L ∩ M**(Intersection)** is a regular language.
   RE_1 = a(a*) and RE_2 = (aa)*
   So, L_1 = { a,aa, aaa, aaaa, ...} (Strings of all possible lengths excluding Null)
   L_2 = { ε, aa, aaaa, aaaaaa,.......} (Strings of even length including Null)
   $L_1 \cap L_2$ = { aa, aaaa, aaaaaa,.......} (Strings of even length excluding Null)
   RE ($L_1 \cap L_2$) = aa(aa)* which is a regular expression itself.

Properties of regular language;

6. If L and M are regular, then so is L − M**(Difference)**.
   Proof: $M - N = M \cap \overline{N}$

7. If L is regular, then so is L^R**(Reversal)**.
   Let, L = {01, 10, 11, 10}
   RE (L) = 01 + 10 + 11 + 10
   L^R = {10, 01, 11, 01}
   RE (L^R) = 01 + 10 + 11 + 10 which is regular