Unit II

DATE: / /

PAGE NO.: 11

Regular Expressions (contd.)

ex. Let V be the lang. of all str. of a's & b's in which either the str. are all b's else there is an a followed by some b's.

$V = \{ \Lambda \ a \ b \ ab \ bb \ abb \ bbb \ abbb \ \dots \}$

RE $b^* + ab^*$

$\because b^* = \Lambda b^*$

$\Lambda b^* + ab^* = (\Lambda + a) b^*$. — distributive law.

$ab = ba$ in algebra

$ab \neq ba$ in formal lang, they are diff. words.

ex. $T = \{ a \ c \ ab \ cb \ abb \ cbb \ \dots \}$

RE - $(a+c) b^* = ab^* + cb^*$.

Sometimes distributive law is not applicable.
Expressions may be distributed but operators cannot.
The star alone can't always be distributed w/o changing meaning of expression.
ex. $(ab)^* \neq a^* b^*$ $\because$ 2 lang. are different.

Multiplication of sets of words –
Def.^n If S & T are sets of strings of letters (finite/infinite) the product set of strings of letters is
ST = { all combinations of a str. from S concatenated with a str. from T in that order }.

ex. $S = \{ a \ aa \ aaa \}$ $T = \{ bb \ bbb \}$

$ST = \{ abb \ abbb \ aabb \ aabbb \ aaabb \ aaabbb \}$.

ex. $P = \{ a \ bb \ bab \}$ $Q = \{ \Lambda \ bbbb \}$

$PQ = \{ a \ bb \ bab \ abbbb \ bbbbbb \ babbbbb \}$.

ex. $L\Lambda = \Lambda L = L$.

DATE: / /
PAGE NO.:

Theorem -

Finite languages are regular - i.e. can be described by RE.
Proof - to make one RE that defines L, turn all words in L into boldface type & insert plus signs between them.

ex: RE that defines lang
$L = \{baa \quad abbba \quad bababa\}$
is $baa + abbba + bababa$

ex. $L = \{aa \quad ab \quad ba \quad bb\}$
RE is $aa + ab + ba + bb$
or $(a+b)(a+b)$
∴ RE need not be unique.

ex. $L = \{\Lambda \quad x \quad xx \quad xxx \quad xxxx\}$
RE - $\Lambda + x + xx + xxx + xxxx$
or $(\Lambda + x)^4$
i.e. $(\Lambda+x)(\Lambda+x)(\Lambda+x)(\Lambda+x)$.

- It is hard to understand a RE -

ex. $(a+b)^*(aa+bb)(a+b)^*$.
Str. of a & b containing a double letter.
What str. do not contain a double letter &
$\Lambda \quad a \quad b \quad ab \quad ba \quad aba \quad bab \quad abab \quad baba \ldots$
expression $(ab)^*$ covers all of these except those that begin with b or end in a. Adding it gives,
$(\Lambda+b)(ab)^*(\Lambda+a)$
combining 2,
$(a+b)^*(aa+bb)(a+b)^* + (\Lambda+b)(ab)^*(\Lambda+a)$
Looking at exprⁿ, it is difficult to tell that it defines all strings.

ex. $E = (a+b)^* a (a+b)^* (a+\Lambda) (a+b)^* a (a+b)^*$.
all words must have at least 2 a's.
break middle + sign,
$\therefore E = (a+b)^* a (a+b)^* a (a+b)^* a (a+b)^*$
$+ (a+b)^* a (a+b)^* \Lambda (a+b)^* a (a+b)^*$ - distributive law.

1st term - words with at least 3 a's.
$(a+b)^* \Lambda (a+b)^* = (a+b)^*$.
$\therefore$ 2nd term reduces to,
$(a+b)^* a (a+b)^* a (a+b)^*$. - all words with at least 2 a's.

$\therefore$ lang. associated with E is the union of all strings that have 3 or more a's with all strings that have 2 or more a's.
But since all str. with 3 or more a's are themselves already str. with 2 or m. a's, this whole lang. is just the 2nd set alone.
Lang. associated with E is no diff. from lang. ass. with
$(a+b)^* a (a+b)^* a (a+b)^*$.

Star is applied to an expression that already has a star in it.
ex. $(a+b^*)^*$. $(aa+ab^*)^*$. $((a+bbba^*)+ba^*b)^*$
Initial star adds nothing to the lang.
$(a+b^*)^* = (a+b)^*$.
also $(a^*)^* = a^*$.
but $(aa+ab^*)^* \neq (aa+ab)^*$.
has abbabb — does not contain abbabb
" " " double b.

ex. $(a^*b^*)^*$ contains all str. of a's & b's.
$= (a+b)^*$.

ex. $b^*(abb^*)^* (\Lambda+a)$. - lang of all words without a double a. Typical words starts with some b's. Then abb^* (a followed by at least one b). Then final a or the last b's as they are.

DATE: / /

PAGE NO.:

Even-Even -

$E = [aa+bb+(ab+ba)(aa+bb)^*(ab+ba)]^*$

words are of 3 types,

1 = aa

2 = bb

3 = $(ab+ba)(aa+bb)^*(ab+ba)$

$E = [\text{type1} + t2 + t3]^*$

Every word of lang. E contains an even No of a's & even No. b's.

2 methods to check this,

① 2 binary flags - a flag & b flag

Every time an a is read, a flag is reversed. (0<->1),
" " b " b "

Initially both are 0 & check they are 0 at end.

② only 1 flag - type 3 flag.

Read 2 letters at a time. If same (type1/type2), don't touch flag. If dont match, we throw (reverse) type 3 flag.

Initially it is 0, whenever it is 1, we are in middle of type 3 factor, when it is 0, we are not. If it is 0 at end, then i/p str. contain even no. of a's & b's, each.

ex. (aa)(ab)(bb)(ba)(ab)·(bb)(bb)(bb)(ab)(ab)(bb)(ba)(aa)
flag is reversed 6 times & ends at 0.

Lang. Even-Even - {Λ aa bb aaaa aabb abab abba baab baba bbaa bbbb aaaaaa ...}

Examples :-

1) $(0+1)^*$ → all strings of 0's & 1's.

2) $(0+1)^* 00 (0+1)^*$ – —"— with at least 2 consecutive 0's.

3) $(1+10)^*$ – —"— beginning with 1 & not having 2 consecu. 0's.

4) $(0+\epsilon)(1+10)^*$ – —"— with no 2 consecu. 0's.

5) $(0+1)^* 011$ – —"— ending in 011.

6) $0^* 1^* 2^*$ – any No. of 0's followed by any. No. of 1's followed by any No. of 2's.

7) $00^* 11^* 22^*$ – —"— with at least one of each symbol.

or $0^+ 1^+ 2^+$.