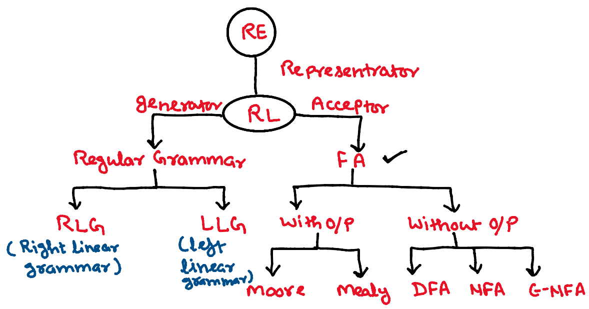
Regular Languages and Regular Expressions

Wednesday, May 5, 2021 10:38 PM

* We have seen many examples such as strings of length $=2, \le 2, \ge 2$, starting with 'a', ending with 'b', divisible by 2, 3, ---- etc. All such problems come under Regular Languages catagory.



* given string is accepted by a m/c (FA) called as acceptor.
* Generator means given a language it will generate all strings in a language
  and acceptor tells whether the given string is there in the language or not.

ex $L_1 = \{a, aa, bb, ----\}$, check whether the string "abb" is present in L_1 or not.
We use acceptor (FA) which produces an output in terms of yes or NO.
But if we want to generate the entire language L_1, then we may use Generator which will generate all the strings for language L_1.

* Generators are nothing but grammars.
* The grammar which generates Regular language is called as Regular Grammar
* Regular languages can be mathematically represented by Regular Expressions

Regular Expressions:

* Representation of the regular languages which are accepted by finite automata
  OR
* If we consider any language that is accepted by FA, then we can represent it using a regular expression.

→ Three basic operations:

i) + (Union operation)
ii) • (concatenation operation)
iii) * (Kleen closure operation)

ϵ → empty string ✓ $(r_1 + r_2)$ ✓
ϕ → empty set ✓ $(r_1 \cdot r_2)$ ✓
r_1^*

→ Regular expressions can be defined as:

a) ϕ, ϵ, $a \in \Sigma$, a, b (Primitive regular expression)
$\{\ \}$ $\{\epsilon\}$, $\{a\}, \{b\}$ ← languages

b) $r_1 + r_2$, $r_1 \cdot r_2$, r_1^* : RE's using three operations on primitive RE's

c) Apply (a) & (b) any number of times and whatever outcome we get will be a RE.

So,
$\phi = \{\ \}$
$\epsilon = \{\epsilon\}$
$a = \{a\}$
$a^* = \{\epsilon, a, aa, aaa, ----\}$
$a^+ = a \cdot a^* = a^* . a = \{a, aa, aaa, ----\}$
$(a+b)^* = \{\epsilon, a, b, aa, ab, ba, bb, ----\}$

$a^0 = \{\epsilon\}$
$a^1 = \{a\}$
$a^2 = \{aa\}$

$\Sigma^+ = \Sigma^* - \{\epsilon\}$

$* = 0$
$* = 1$
$* = 2$

$(a+b)^* = \{\epsilon, a, b, aa\ ab, ba, bb$
$------\}$
$* = 3$ ---

ex $L_1 = \{$ language of all strings whose length is exactly 2 $\}$

finite → $L_1 = \{aa, ab, bb, ba\}$, $\Sigma = \{a, b\}$.

$aa + ab + bb + ba$
$\Rightarrow a(a+b) + b(a+b)$
$\Rightarrow \boxed{(a+b) \cdot (a+b)} \Rightarrow (a+b)^2$

* If the language is finite then there is definitely a FA as well as RE is also possible
* If the language is finite then the corresponding RE will be the union of all the string of that language

→ $aa + ab + ba + bb$
$\Rightarrow a(a+b) + b(a+b)$
$\Rightarrow (a+b)(a+b) \Rightarrow$ RE for set of strings whose length is exactly 2

$(a+b)\ (a+b)$
$(a \text{ or } b) \cdot a \text{ or } b$
$a \cdot a = aa$
$b \cdot a = ba$
bb bb

Similarly, $(a+b)(a+b)(a+b) \Rightarrow$ RE " " " " " " " 3
& $(a+b)(a+b)(a+b)(a+b) \Rightarrow$ RE — ——— - - — — — 4, and so on.

$\Rightarrow$ $(a+b)(a+b) \Rightarrow$ RE for set of strings whose length is exactly 2

similarly, $(a+b)(a+b)(a+b) \Rightarrow$ RE " " " " " " " 3

& $(a+b)(a+b)(a+b)(a+b) \Rightarrow$ RE — ——— – – — — – 4, and so on.

$a \cdot a = aa$
$b \cdot a = ba$
bb $\quad$ bb

ex $L_1 = \{$ set of all strings whose length is atleast 2 $\}$ $\Rightarrow |w| \geq 2$

$L_1 = \{ aa, ab, ba, bb, aaa, abaa, \cdots \} \rightarrow$ infinite

2, 3, 4, ---

$\rightarrow (a+b)(a+b)(a+b)^*$

we know that $(a+b)(a+b) \rightarrow$ RE for set of all strings whose length is exactly 2

$* = 0 \quad |w| = 2$

$* = 1 \quad |w| = 3$

so $(a+b)(a+b)(a+b)^* \Rightarrow$ RE for length 2, 3, 4 ----

$* = 0$, then $(a+b)(a+b) \Rightarrow |w| = 2$

$* = 1$, $|w| = 3$

$* = 2$, $|w| = 4$

Identities of Regular Expressions:

1.) $\emptyset + R = R + \emptyset = R$

2.) $\emptyset . R = R . \emptyset = \emptyset$

3.) $\epsilon . R = R . \epsilon = R$

4.) $\epsilon^* = \epsilon$

5.) $\emptyset^* = \epsilon$

6.) $\epsilon + RR^* = R^*R + \epsilon = R^*$

7.) $(a+b)^* = (a^*+b^*)^*$

$= (a^* b^*)^*$

$= (a^*+b)^*$

$= (a+b^*)^*$

$= a^*(ba^*)^*$

$= b^*(ab^*)^*$

8.) $R + R = R$

9.) $R^* R^* = R^*$

10.) $RR^* = R^*R$

11.) $(R^*)^* = R^*$

12.) $(P+Q)R = PR + QR$,

$R(P+Q) = RP + RQ$

$R^+ = R . R^*$ $\qquad$ $R^* = R^+ \cup \epsilon$

$a . \{ \epsilon, a, aa, aaa \cdots \}$

$R^+ = \{a, aa\ aaa \cdots\} \cup \{\epsilon\}$

$= \{\epsilon, a\ aa, aaa \cdots\} = R^*$

ARDEN'S THEOREM:

Let P and Q be two regular expressions over Σ. If P does not contain ϵ, then the following equation in R, viz.

$$R = Q + RP$$

has one and only one solution given by

$$\boxed{R = QP^*}$$

Proof: for given solution $R = QP^*$, lets put it in $R = Q + RP$

$\Rightarrow Q + (QP^*)P = Q(\epsilon + P^*P)$

$= QP^*$ by Prop. 6

Now, to provide uniqueness of the solution: Propagate the RHS in a given equation

$$Q + RP = Q + (Q+RP)P$$
$$= Q + QP + RPP$$
$$= Q + QP + RP^2$$
$$= Q + QP + QP^2 + \cdots + QP^i + RP^{i+1}$$
$$= Q(\epsilon + P + P^2 + \cdots P^i) + RP^{i+1} \quad \text{for } i \geq 0$$

let us consider a string w of length i, $|w| = i$

then $w \in \{ Q(\epsilon + P + P^2 + \cdots + P^i) + RP^{i+1} \}$

In the statement P does not contain ϵ & RP^{i+1} has no string of length less than $i+1$,

therefore, $w \notin RP^{i+1}$ as $|w| = i$

$\Rightarrow w \in Q(\epsilon + P + P^2 + \cdots + P^i)$

$\Rightarrow w \in QP^*$