Regular Expression (Lecture 2)

A Regular Expression can be recursively defined as follows:

1. ε is a Regular Expression indicates the language containing an empty string. ($L(\varepsilon) = \{\varepsilon\}$ one string, with no symbols).
2. φ is a Regular Expression denoting an empty language. ($L(\varphi) = \{\ \}$ no strings).
3. x is a Regular Expression where $L=\{x\}$ one string, with one symbol x.
4. If X is a Regular Expression denoting the language L(X) and Y is a Regular Expression denoting the language L(Y), then:
   a) X + Y is a Regular Expression corresponding to the language L(X) U L(Y) where L(X+Y) = L(X) U L(Y).
   b) X . Y is a Regular Expression corresponding to the language L(X) . L(Y) where L (X.Y) = L(X) . L(Y)
   c) R* is a Regular Expression corresponding to the language L(R*) where L(R*) = (L(R))*
5. If we apply any of the rules several times from 1 to 5, they are Regular Expressions.
   - $\mathbf{L(R_1 \cup R_2) = L(R_1) \cup L(R_2)}$
   - $\mathbf{L(R_1 \circ R_2) = L(R_1) \circ L(R_2)}$
   - $\mathbf{L(R_1^*) = (L(R_1))^*}$

Example

Expression $((0 \cup 1)\,\varepsilon)^* \cup 0$ denotes language $\{0, 1\}^* \cup \{0\} = \{0, 1\}^*$, all strings.

Example

$(0 \cup 1)^*\ 111\ (0 \cup 1)^*$ denotes $\{0, 1\}^*\ \{111\}\ \{0, 1\}^*$, all strings with substring 111.

Example

L = strings over {0, 1} with odd number of 1s. $0^*\ 1\ 0^*\ (0^*\ 1\ 0^*\ 1\ 0^*)^*$

Example

L = strings with substring 01 or 10.

$[(0 \cup 1)^*\ 01\ (0 \cup 11)^*] \cup [(0 \cup 1)^*\ 10\ (0 \cup 1)^*]$

Abbreviate (writing Σ for $(0 \cup 1)$):

$\Sigma^* \, 01 \, \Sigma^* \cup \Sigma^* \, 10 \, \Sigma^*$

Example

L = strings with substring 01 or 10.

$(0 \cup 1)^* \, 01 \, (0 \cup 1)^* \cup (0 \cup 1)^* \, 10 \, (0 \cup 1)^*$

Abbreviate:

$\Sigma^* \, 01 \, \Sigma^* \cup \Sigma^* \, 10 \, \Sigma^*$

Example

L = strings with neither substring 01 or 10.

– Can't write complement.

– But can write: $0^* \cup 1^*$.

Example

L = strings with no more than two consecutive 0s or two consecutive 1s

– Would be easy if we could write complement.

$(\varepsilon \cup 1 \cup 11).\ ((0 \cup 00).\ (1 \cup 11))^*\ (\varepsilon \cup 0 \cup 00)$

– Alternate one or two of each.

- Regular expressions commonly used to specify syntax.
- For (portions of) programming languages –Editors – Command languages like UNIX shell

Example

Decimal numbers $DD^*.\ D^* \cup D^*.\ D\ D^*$,

Where D is the alphabet {0, …, 9 } Need a digit either before or after the decimal point.

Some RE Examples

Regular Expression	**Regular Set**
(0+10*)	L= { 0, 1, 10, 100, 1000, 10000, … }
(0*10*)	L={1, 01, 10, 010, 0010, …}
(0+ε).(1+ ε)	L= {ε, 0, 1, 01}
$(a+b)^*$	Set of strings of a's and b's of any length including the null string. So L= { ε, a, b, aa , ab , bb , ba, aaa…….}
$(a+b)^*abb$	Set of strings of a's and b's ending with the string abb. So L = {abb, aabb, babb, aaabb, ababb, …………..}
$(11)^*$	Set consisting of even number of 1's including empty string, So L= {ε, 11, 1111, 111111, ……….}
$(aa)^*(bb)^*b$	Set of strings consisting of even number of a's followed by odd number of b's , So L= {b, aab, aabbb, aabbbbb, aaaab, aaaabbb,……..}
$(aa + ab + ba + bb)^*$	String of a's and b's of even length can be obtained by concatenating any combination of the strings aa, ab, ba and bb including null, so L= {e,aa, ab, ba, bb, aaab, aaba, …………..}

– The languages that are associated with these regular expressions are called regular languages.

Example

Consider the language L Where L= {Λ x xx xxx …} by using star notation, we may write L=language(x^*).

Since x^* is any string of x's (including Λ).

Example

If we have the alphabet Σ= {a,b} And L={a ab abb abbb abbbb …} Then L=language(ab^*)

Example

$(ab)^*$= Λ or ab or abab or ababab or abababab or ….

Example

L_1=language (xx^*)

The language L_1 can be defined by any of the expressions:

xx^* or x^+ or xx^*x^* or x^*xx^* or x^+x^* or x^*x^+ or $x^*x^*x^*xx^*$ … Remember x^* can always be Λ.

Example

Language (ab^*a) = {aa aba abba abbba abbbba …}

Example

Language (a^*b^*) = {Λ a b aa ab bb aaa aab abb bbb …} ba and aba are not in this language so $a^*b^* \neq (ab)^*$

Example

The following expressions both define the language L_2= {x^{odd}}: $x\ (xx)^*$ or $(xx)^*x$ But the expression x^*xx^* does not since it includes the word (xx) x(x).

Example

Consider the language T defined over the alphabet Σ= {a,b,c} T={a c ab cb abb cbb abbb cbbb abbbb cbbbb …}

Then T=language ($(a+c)\ b^*$) T=language (either a or c then some b's)

Example

Consider a finite language L that contains all the strings of a's and b's of length exactly three.

L= {aaa aab aba abb baa bab bba bbb} L=language ((a+b) (a+b) (a+b)) L=language ((a+b) 3)

Note: from the alphabet Σ={a,b} , if we want to refer to the set of all possible strings of a's or b's of any length (including Λ) we could write $(a+b)^*$

Example

We can describe all words that begins with a and end with b with the expression *a* $(a+b)^*b$ which mean *a (arbitrary string) b*

Example

If we have the expression $(a+b)^*a\ (a+b)^*$ then the word abbaab can be considerd to be of this form in three ways: (Λ) a (bbaab) or (abb) a (ab) or (abba) a (b)

Example

$(a+b)^*a\ (a+b)^*a\ (a+b)^*$ = (some beginning) (the first important a) (some middle) (the second important a) (some end)

Another expressions that denote all the words with at least two a's are: $b^*ab^*a\ (a+b)^*$, $(a+b)^*ab^*ab^*$, $b^*a\ (a+b)^*ab^*$

Then we could write:

=language ((a+b)*a(a+b)*a(a+b)*)

=language(b*ab*a(a+b)*)

=language((a+b)*ab*ab*)

=language(b*a(a+b)*ab*)

=all words with at least two a's.

Note: two regular expressions are equivalent if they describe the same language.

Example

If we want all the words with exactly two a's, we could use the expression: $b^*ab^*ab^*$ which describe such words as aab, baba, bbbabbabbbb,…

Example

The language of all words that have at least one a and at least one b is

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

Note: $(a+b)^*b\ (a+b)^*a\ (a+b)^* \neq bb^*aa^*$ since the left includes the word aba, which the expression on the right side does not.

Note:

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

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

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

Note: usually when we employ the star operation we are defining an infinite language. We can represent a finite language by using the plus alone.

Example

L= {abba baaa bbbb}

L=language (abba + baaa + bbbb)

Example

L= {Λ a aa bbb}

L=language(Λ + a + aa + bbb)

Example

L= {Λ a b ab bb abb bbb abbb bbbb …}

We can define L by using the expression $b^* + ab^*$

Definition

The set of regular expressions is defined by the following rules:

Rule1: every letter of Σ can be made into a regular expression, Λ is a regular expression.

Rule2: if R_1 and R_2 are regular expressions, then so are: (R_1) R_1R_2 R_1+R_2 R_1^*.

Rule3: nothing else is a regular expression. Remember that $R_1^+=R_1R_1^*$

Definition

If S and T are sets of strings of letters (whether they are finite or infinite sets), we define the product set of strings of letters to be: ST= {all combination of a string from S concatenated with a string from T}

Example

If S= {a aa aaa} T= {bb bbb}

Then ST= {abb abbb aabb aabbb aaabb aaabbb} (a+aa+aaa) (bb+bbb) =abb+abbb+ aabb+aabbb+aaabb+aaabbb)

Example

If P= {a bb bab} Q= {Λ bbbb} Then PQ= {a bb bab abbbb bbbbbb babbbbb}

$(a+bb+bab)(\Lambda+bbbb)=a+bb+bab+ab^4+b^6+bab^5$

Example

If M= {Λ x xx} N= {Λ y yy yyy yyyy …}

Then MN= {Λ y yy yyy yyyy …

x xy xyy xyyy xyyyy …

xx xxy xxyy xxyyy xxyyyy …}

Using regular expression we could write: $(\Lambda+x+xx)(y^*)=y^*+xy^*+xxy^*$

Definition

The following rules define the language associated with any regular expression.

Rule1: the language associated with the regular expression that is just a single letter is that one-letter word alone and the language associated with Λ is just{Λ}, a one-word language.

Rule2: if R_1 is regular expression associated with the language L_1 and R_2 is regular expression associated with the language L_2 then:

i. The regular expression (R_1) (R_2) is associated with the language L_1 times L_2.

$$\text{Language }(R_1R_2)=L_1L_2$$

ii. The regular expression R_1+R_2 is associated with the language formed by the union of the sets L_1 and L_2.

$$\text{Language }(R_1+R_2)=L_1+L_2$$

iii. The language associated with the regular expression $(R_1)^*$ is L_1^*, the kleene closure of the set L_1 as a set of words.

$$\text{Language }(R_1)^*=L_1^*$$

Example

$L= \{baa\ abba\ bababa\}$

The regular expression for this language is (baa+abba+bababa)

Example

$L= \{\Lambda\ x\ xx\ xxx\ xxxx\ xxxxx\}$

The regular expression for this language is $(\Lambda+x+xx+xxx+xxxx+xxxxx)$

$= (\Lambda+x)^5$

Example

L= language $((a+b)^*(aa+bb)\ (a+b)^*)$

= (arbitrary) (double letter) (arbitrary)

$\{\Lambda\ a\ b\ ab\ ba\ aba\ bab\ abab\ baba\ \ldots\}$ these words are not included in L but they included by the regular expression: $(\Lambda+b)\ (ab)^*(\Lambda+a)$

Example

$E=(a+b)^*a(a+b)^*(a+\Lambda)(a+b)^*a(a+b)^*$
$E=(a+b)^*a(a+b)^*a(a+b)^*a(a+b)^*+(a+b)^*a(a+b)^*\Lambda(a+b)^*a(a+b)^*$

We have $(a+b)^*\Lambda\ (a+b)^*= (a+b)^*$

Then: $E= (a+b)^*a\ (a+b)^*a\ (a+b)^*a\ (a+b)^*+ (a+b)^*a\ (a+b)^*a\ (a+b)^*$

The language associated with E is not different from the language associated with: $(a+b)^*a\ (a+b)^*a\ (a+b)^*$

Note: $(a+b^*)^*= (a+b)^*$ $(a^*)^*=a^*$ $(aa+ab^*)^*\neq (aa+ab)^*$ $(a^*b^*)^*= (a+b)^*$

Example

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

Even-even= $\{\Lambda\ aa\ bb\ aabb\ abab\ abba\ baab\ baba\ bbaa\ aaaabb\ aaabab\ldots\}$