
بسم الله الرحمن الرحيم

Introduction

Dr. Nasir Jalal PhD (Computer Science)

Lecturer

CS & IT Department, CUVAS Bahawalpur

Course Detail

BS CS 5th Semester Fall 2022-2026

Course: Theory of Programming Languages

Course Code: CS IT 507

Lecture: 8 (26-09-2024)

Outline Lecture 8

- Regular Expression
- Groups and Sub Expression

Regular Expression

- Symbol “a” mean must be “a”
- Symbol “b” mean must be “b”
- (a+b) mean must be “a” or “b”
- (a+b)* mean none, a or b any number of times (Recursively)
- (a+b)* or (a*b*)* are same

Note

Consider the language

L={aaa, aab, aba, abb, baa, bab, bba, bbb} that may be expressed by a Regular Expression aaa+aab+aba+abb+baa+bab+bba+ bbb, which is equivalent to

(a+b)(a+b)(a+b).

Equivalent Regular Expressions

Definition:

Two regular expressions are said to be equivalent if they generate the same language.

Example:

Consider the following regular expressions

$r_1 = (a + b)^* (aa + bb)$

$r_2 = (a + b)^* aa + (a + b)^* bb$ then

both regular expressions define the language of strings **ending in aa or bb**.

Task

Difference between R.E a* + b* and R.E (a+b)*

Task

Difference between R.E a* + b* and R.E (a+b)*

Solution!

$a^* + b^*$ does not generate any string of concatenation of a and b

$(a+b)^*$ generates concatenation of strings

Groups and Ranges

A group is a specific set of characters that will match any character inside it. For instance, *[abcde]* is a group that would match any of those five letters.

Character	Full Name	Regex Type	Description
[ ]	Square Brackets	Grouping	Creates a character group or range.
( )	Parentheses	Grouping	Creates a sequence or sub-expression.

Groups and Ranges

You can negate a group by adding a caret as the first character.

- *[^a-zA-Z]* will match any non-letter character

[^0-9] will match any non-numeric character.

Sub-expression in Regular Expression:

- Parentheses () are used to create **sub-expressions** or **capture groups** in regular expressions. This allows you to group multiple tokens together and extract the matched part.
- RE: (\d{3})-(\d{4})

For string "Phone: 123-4567“

it will match 123-4567, and will capture:

123 in the first capture group.

4567 in the second capture group.

Quantifiers

Quantifiers allow for some flexibility in matching as they define the number of times a character, pattern, or group appears in a regex match.

Character	Full Name	Regex Type	Description
?	Question Mark	Quantifier	Matches zero or one preceding character.
*	Asterisk	Quantifier	Matches zero or more preceding characters.
+	Plus Sign	Quantifier	Matches one or more preceding characters.
{ }	Curly Braces	Quantifier	Creates a specific numerical quantifier range.
{5}	Curly Braces	Quantifier	Matches exactly five characters.
{2,5}	Curly Braces	Quantifier	Matches between two and five characters.
{2,}	Curly Braces	Quantifier	Matches two or more characters.

Quantifiers +

- you might want to match as many characters in the group *[a-zA-Z]* as possible, as long as at least one letter is present.
- you can use the plus sign character after the group to match one or more of the preceding characters.

  RE is **[a-zA-A]+**
- at least one digit but allows for an infinite number of digits.

[0-9]+

Quantifiers ?

The question mark is useful for optional characters. For example, write the below expression if you want to match the names Ashle, Ashlee, and Ashley.

Ashle[ey]?

The quantifier doesn't only have to follow a group; it can follow a single character as well.

Metacharacters

Metacharacters, also known as shorthand, are additional special characters that replace longer BRE expressions.

Metacharacter	Replaced Expression	Description
\d	[0-9]	Matches a single digit.
\D	[^0-9]	Matches a single non-digit.
\s	[\t\r\n\f]	Matches whitespace (space, tab, return, newline, or fullstop).
\S	{^ \t\r\n\f]	Matches non-whitespace.
\w	[a-zA-Z0-9_]	Matches a single word character (including digits and underscore).
\W	[^a-zA-Z0-9_]	Matches a non-word character.
.	[a-zA-Z0-9_-=+'";:….]	Matches any single character.

Metacharacters

Write RE for apartment numbers such that apartment numbers will always have one letter and one to three digits, surrounded by whitespace.

- **\s[a-zA-Z]\d+\s**

This expression matches like so:

- **Whitespace**: *\s*
- **A single letter (upper or lower case)**: *[a-zA-z]*
- **One or more digits**: *\d+*
- **Whitespace**: *\s*

THANKS