Reg(ular)?Ex(pressions)?

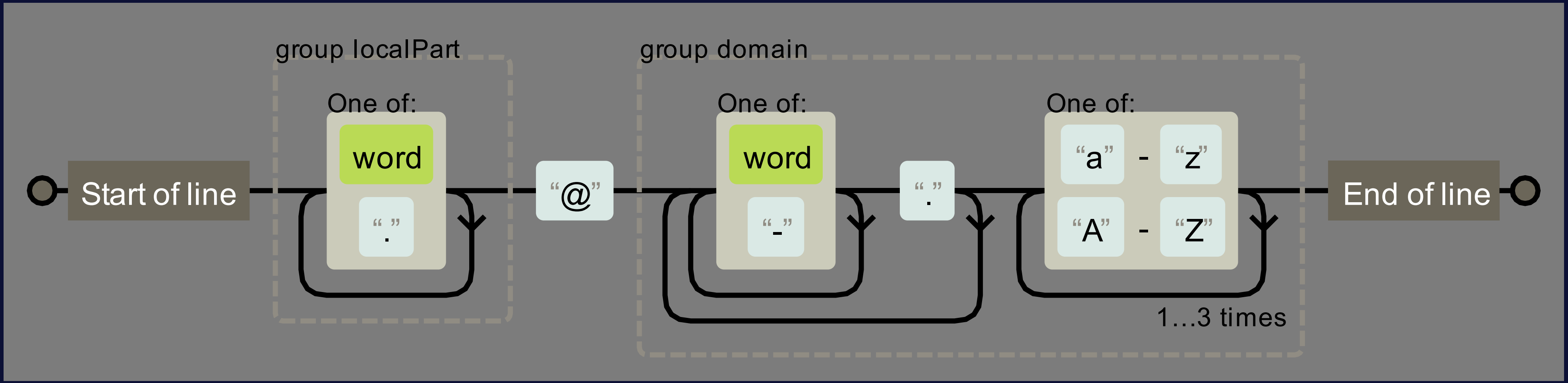
By Mohammad Omidvar T. – Spring 2019

What is a Regex?

- A simple but powerful pattern that is used for **searching and validating strings**.
- Has its own syntax
- Almost every programming language supports it. But they may have different flavors.

Example

- Validating an email address:
- `^(?<localPart>[\w.]+)@(?<domain>(?:[\w\-]+\.)+[a-zA-Z]{2,4})$`



Let’s start

Simple ‘is’ Finder

Regex	`is`

Text	`Today is Sunday.`
	`Teaching assistants`
	`Analysis`

\b
Word boundary

Match the position at the beginning or end of any word

Simple ‘is’ Finder (to be verb)

Regex	`\bis\b`
Text	Today is Sunday. Teaching assistants Analysis

Character Classes

\d
Digit

Matches any digit character (0 to 9)

\w
Word

Matches any alphanumeric character
(a to z, A to Z, digits, _)

\s

Space

Match any whitespace character
(space, new line, tab, …)

. (dot)
Any

Match any character except newline

[classes]

Matches any single character in the list

Simple Digit Finder

Regex	\d
Text	Hi 1234 a1b2c3

A Capital Letter at Start of a Word

Regex	`\b[A-Z]`
Text	Hi there! Hello World!

Simple Phone Number Validator

Regex	\d\d\d\d\d\d\d\d\d\d\d
Text	09151234567 12345678901

Simple Cellphone Number Validator

Regex	`09\d\d\d\d\d\d\d\d\d`
Text	09151234567 12345678901

Too many \ds!

Quantifiers

*

Zero or more

+

One or more

?

Zero or one

{n}

n times

{n,}

At least n times

{n,m}

At least n times, at most m times

aaa...aaab!

Regex	a+b
Text	aaaaab ab b

Find Numbers in Text

Regex	`\d+`
Text	`Hi there! 12 + 2 = 14`

Simple Cellphone Number Validator

Regex	`09\d{9}`
Text	09151234567 12345678901

Extending the Regex

- What if our phone number has dashes?
- 021-12345678
- 0915-12345678

Regex	`\d{3,4}-?\d+`

Text	`021123`
	`021-12345678`
	`021--123456`
	`0931-1234567`

Words Starting with a Vowel

Regex	`\b[aeiouAEIOU]\w+\b`
Text	Apple internet basket

Java Identifier Detector

Regex	`[a-zA-Z_$][\w$]*`

Text	`scanner`
	`player1Score`
	`towerHeight`
	`23test`

Try to write a regex for:

- Matching this format:
  - HH:mm -> HH:mm
  - 12:30 -> 18:15

Words Preceded by Indefinite Articles (a)

Regex	`a \w+`

Text	`a book`
	`test`
	`an egg`

But some words start with vowels and need ‘an’…

|
Alternatives

One of two patterns will be accepted

Words Preceded by Indefinite Articles (a, an)

Regex	`(a|an) \w+`

Text	`a book`
	`test`
	`an egg`
	`an basket`

But ‘an’ should not come before a consonant (almost all cases)

Negation

Negatives

Class	Negative	Description
\d	\D	Any non-digit character
\w	\W	Any non-alphanumeric character
\s	\S	Any character that is not whitespace
[classes]	[^classes]	Anything except these classes

Words Preceded by Indefinite Articles (a, an)

Regex	`(a [^aeiou\d\W]\w+|an [aeiou]\w+)`

Text	`a book`
	`test`
	`an egg`
	`an basket`

Now it works better!

You can use it to find grammar problems!

Groups

What are a group and its usages?

- A pattern and a part of the whole pattern
- Retrieving part of a matched string
- Repeating a pattern
- Making references inside text

IP Validator (without limitation)

- Consider $\boxed{\text{192.}}$168.1.1

Repeating part: `(\d+\.)`

Regex	`(\d+\.){3}\d+`
Text	192.168.1.1 100.120.112.12 100000.1.0.400 10.500.3

Hex Color Validator

- Consider #FFF or #1D2F3C or #DFFF or #FF000000
- The length must be 3 or 6, or 4 or 8

Regex	`#(([0-9a-fA-F]{3}){1,2}|([0-9a-fA-F]{4}){1,2})`
Text	#121212 #ABC #ABCD0 #12A36 #F1C9 #FFAA1414

Function Call Detector

- Consider

```
myObject.myField.myMethod(val1, val2, val3);
```

Repeating part: `identifier.`

Repeating part: `identifier,`

Regex	`(idntfr\.)*idntfr\((idntfr,)*idntfr\);`

Text	`System.out.println(myText);`
	`repeat(myText, count);`

Replace ‘idntfr’ with `[a-zA-Z_$][\w$]*`

```
Full regex: ([a-zA-Z_$][\w$]*\s*\.\s*)*[a-zA-Z_$][\w$]*\s*\(([a-zA-Z_$][\w$]*\s*,\s*)*[a-zA-Z_$][\w$]*\s*\)\s*;
```

Now, Try to extend your pattern

- You can pass different things as parameters
  - values (numbers, strings, …)
  - variables (local fields, class members)
  - …

IP Validator

- Now limit your pattern to validate an ip with consideration of numbers <= 255

Regex	`((25[0-5]|((2[0-4]|[01]\d)\d)|\d{1,2})\.){3}(25[0-5]|((2[0-4]|[01]\d)\d)|\d{1,2})`
Text	192.168.1.1 255.255.0.0 134.266.0.1

Retrieve Part of Text

- Try to find a pattern for parsing a date:
- Regex: `(\d{2}|\d{4})/(\d{1,2})/(\d{1,2})`
- Text: `2019/1/10`

Anchors

^

Match the beginning of the string (line)

$

Match the end of the string (line)

\b
Word boundary

Match the position at the beginning or end of any word

Matching Starting Whitespace

Regex	^\s*
Text	🟥test

Http URL Matcher

Regex	^(https?:\/\/)?(([\w\-]+\.)+[a-zA-Z]{2,4})(:\d+)?\/?([\w\-._?,'\/\\+&;%$#=~]*)$
Text	quera.ir https://quera.ir/course/2770/ https://www.google.com/search?q=hi&ie=&oe= www.google

Advanced Regex

(?<name>exp)

Named group

(?:exp)

Non-capturing group

Now, back to the email example

- `^(?<localPart>[\w.]+)@(?<domain>(?:[\w\-]+\.)+[a-zA-Z]{2,4})$`
- `Is it that much complicated?!`

Quantifier?

Lazy quantifier (as few as possible)

Greedy vs Lazy

Regex	Text
(A+?)(A*)	AAAAAA
(A+)(A*)	AAAAAA

\n (slash number)

Back reference to the nth group

HTML Tag Matcher

Regex	^<(\w+)(?:>(.*)<\/\1>|([^>]+)\/>)$
Text	<html><body>Hi</body></html>
	<Button android:text = “Hi”/>

Lookarounds

(?=pattern)

Lookahead

Gerund Matcher

Regex	`\b\w+(?=ing\b)`
Text	doing winger going

- Attention: Lookarounds work like anchors. They just match positions and are not capturing groups.

(?<=pattern)

Lookbehind

(?!pattern)

Negative Lookahead

(?<!pattern)

Negative Lookbehind

[class1&&class2]

Class Intersection

(Java flavor)

Words Starting with a Consonant

Regex	`\b[a-zA-Z&&[^aeiouAEIOU]]\w+`
Text	Apple snake test