COMPSCI 221: Programming Assignment 4 Regex

Due on Tuesday, March 21, 2017

Dr. Leyk

Robert Quan

Problem 1

Abstract
In this assignment, we begin to use the searching language Regex to search for strings or keywords withing a data set. Regex is a very powerful and useful language to know. In this assignment I have written the correct Regex expressions to find the keys inside a string that was assigned to us.

3)
a) *What is stored in "matches"?*

The variable matches contains all the matches keys found from the string by using the correct regex notation. It is an array that contains all the found instances of the keys.

b) *What does* \d *"mean?*

"\d" is Regex code for a digit character. In our case, the code will find the first instance of one digit character. If we use the input string *"I would like the number 98"" to be found and printed, thanks."*, then our regex match will be *"9"*.

c) *What does* "\d *"mean?*

To correctly modify my expression for the Regex, I used the following code:

```
regex pattern{R"((\d\d)(?:.+)(thanks))"};
```

Figure 1: Code used for Regex pattern, part 1.

This displays the following output on the terminal:

```
98 to be found and printed, thanks
98
thanks
```

Figure 2: Output on term

It is interesting to note that three strings were grabbed during this process, but the second and third string do match the required strings.

Problem 2

a) *What does* "\s\S"*mean?*
*backslash*s is the Regex code for a whitespace character to be found and *backslash*S denotes any character that is NOT a whitespace. An example of this would be any space and then the first letter of the next word would be taken as an entry to our matches array.

b) *What is stored inside matches[0]?*
Our matches array will find any matching regex code to our string. Inside of matches[0], while using the given regex statment, we have " < *title* >This is a title< /*title* > ". The matching is case sensitive. This first Capturing Group matches the characters < *title* > literally and any character up to < \title> in our first capturing group.

c) *Why is matches[1] different?*
Matches[1] contains the string "This is a title". We can see that it excluded the < *title* > and < /*title* > this is because of the sub group that we put in ([\s\S]+). The second capturing group has a Quantifier that matches between one and unlimited times.

d) *Modify the Regex code*
After modifying the Regex pattern to retrieve only the items indide of the header tag but not inside the title tag, I ended up writing this pattern:

```
regex pattern{R"(<head>((?:(?!<title>).)*)(<title>[^<]+</title>)?([\s\S]*)</head>)"};
```

Figure 3: Code used for Regex pattern, part 2.

This displays the following output on the terminal:

```
Wow such a header
So top
```

Figure 4: Output on term

An interesting note on this pattern is that I had to print out the second and fourth match on out matches array. This is because I used a Capturing group which had a total of 4 group matching.

Problem 3

a) *Write a program using regex that will go through the text file and print out the file name of every hyperlinked powerpoint*

For this problem, I created a fstream object that read in the stroustrup.txt file. After writing the correct pattern to search for, I found the output by setting the matches into a string object and then checking whether the size was bigger than 0 characters in length. This successful printed out the filenames I needed. The code and output are the following:

```
ifstream out("stroustrup.txt");     //input txt
regex pattern("<a href=\"(.*ppt)\">");      //Pa
```

Figure 5: Pattern for stroustrup.txt

```
[rquan@fedoration A4]$ ./a.out
1-2_programming.ppt
3_types.ppt
4_computation.ppt
5_errors.ppt
6_writing.ppt
7_completing.ppt
8_functions.ppt
9_classes.ppt
10_iostreams.ppt
11_custom_io.ppt
12_display.ppt
13_graph_classes.ppt
14_class_design.ppt
15_graphing.ppt
16_GUI.ppt
17_free_store.ppt
18_arrays.ppt
19_vector.ppt
20_containers.ppt
21_algorithms.ppt
22_ideals.ppt
23_text.ppt
24_numerics.ppt
25_embedded.ppt
26_testing.ppt
27_C.ppt
Review_for_Final.ppt
```

Figure 6: Output on term