CS 230
Programming Languages

10 / 03 / 2022

Instructor: Michael Eckmann

Today's Topics

- Questions? / Comments?
- More Regular Expressions (Regexs)
- Quick review of what we learned so far
- More special characters and their meaning in Regular Expressions
- Code examples in Java

Regex

- Quick review of some regular expression material from last time
- Character classes using [ ]
- Use of ^ in a character class
- Use of – in a character class
- Predefined character classes like: . \d \s \D \S
- Recall that when a regular expression tries to match it matches as early as possible in the string

Regex

- Metacharacters {}[]()^$.|*+?\ have special meaning
- | alternation character (acts like a logical or)
- Repetitions

  ? - 0 or 1 time

  * - 0 or more times

  + - 1 or more times

  { } - range (i.e. min and max), at least or exactly

  – These repetition signifiers are placed to the right of the thing to be repeated. They are greedy --- they match as many as possible while still allowing the whole regex to match.

Can look here for meanings of regex characters for Java:

https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html

Regex

- Example:
- Match 1 or more letters (of either case) or spaces, greedily, followed by an e
- “[a-zA-Z]+e” --- note that “ ” inside the character class after Z represents a space

- Input text: “Skidmore College is the premier school in the northeast.”

- With greedy repetition, it tries to match as much of the string as possible and the moves on to try to match the e and backtracks as necessary.
- So, let's walk through the process.

Regex

- A ? after any of the repetition quantifiers says to make it reluctant (opposite of greedy).
- Example:
- Match 1 or more letters (of either case) or spaces, reluctantly, followed by an e
- "[a-zA-Z]+?e" --- note that ? after + makes + reluctant

- Input text: "Skidmore College is the premier school in the northeast."

- One thing to note, with all the examples so far, that the match tried to be as early in the input text as possible, that takes precedence.

Regex

- Parentheses around parts of the regex cause groupings
  - That is, if the regex matches, we don't only have access to the the portion of the text that is the entire match, but also what matched to each of the groupings.
  - e.g. regex: “([0-9]+)\s*([a-zA-Z]+)”
  - If match happens, the digits will go in group 1, the letters in group 2.
  - Note: +'s inside parens, what if + outside of parens?
  - Additionally, vertical bars are often used within parentheses to act as an OR e.g. “(cat|bat|hat)” matches cat, bat OR hat
  - How are these similar and different to character classes?

Regex

- Let's look here for the 4 principles (what takes precedence when attempting to match) that are followed:

http://www.cs.rit.edu/~afb/20013/plc/perl5/doc/perlretut.html

0. Match as early in the input text as possible
1. leftmost alternation that can match, matches
2. quantifiers are greedy
3. if multiple greedy quantifiers, leftmost quantifiers in the regex take precedence for greediness