Background on RegEx TIFFANY

Regular expressions are present in a vast amount of scripting languages (Perl, Java, Python, et cetera) and serve as a powerful programming tool using algebraic notation (Medeiros, Mascarenhas, & Ierusalimschy, 2014) (Campeanu & Santean, 2009). The syntax of regular expressions specify a pattern a user is searching for in a string or a full string against a list of strings (Medeiros, Mascarenhas, & Ierusalimschy, 2014). A modification of regular expressions called the extended regular expressions with back-references (regex), creates a means of creating expressions for patterns (repetitions) by performing a usual pattern match, but then “backtracking when a particular path through the expression makes the match fail”. (Schmid, 2013) (Medeiros, Mascarenhas, & Ierusalimschy, 2014). This backtracking or back-referencing refers to the revisiting of the earlier subexpression when a match fail occurs, to provide a more accurate pattern search throughout a string (Schmid, 2013). “For example, r:=( 1 (a|b)\*)1c\1 is a regex, where \1 is a backreference to the referenced subexpression in between the parentheses (1 and)1 (Schmid, 2013). In this example r denotes the set within the string, a|b are alterations of the search pattern, and c denotes the frequency (Schmid, 2013). Ad-hoc optimizations are used to reduce the amount of backtracking of regex implementation, which in turn decreases the running time of the pattern search (Medeiros, Mascarenhas, & Ierusalimschy, 2014).

When implementing regex as a programming tool, there are specifics in syntax which is dependent on the programming language. In regards to our GFinder algorithm, Java programming language was used. When regex is introduced within a class in Java, the pattern is first specified as a string, then compiled into an instance of the class (Oracle, 2014). The pattern is then “used to create a Matcher object that can match arbitrary character sequences against the regular expression” (Oracle, 2014). Figure [#] is an example of a typical regex invocation sequence.

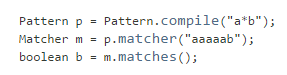


Figure [#]: An example of a typical regex invocation sequence.

Oracle (2014). Oracle: Class pattern. Retrieved from http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html

Reference:

Campeanu C., & Santean, N. (2009). On the intersection of regex languages with regular expression. *Theoretical Computer Science*, 410, 2336-2344. <http://dx.doi.org/10.1016/j.tcs.2009.02.022>

Schmid, M.L. (2013). Inside the class of regex languages. International Journal of Foundations of Computer Science, 24(7), 1117-1134. Doi: 10.1142/S0129054113400340.

Medeiros, S., Mascarenhas, F., & Ierusalimschy, R. (2014). From regrexes to parsing expression grammars. Science of Computer Programming, 93, 3-18. Doi: 10.1016/j.scico.2012.11.006.

Oracle (2014). Oracle: Class pattern. Retrieved from http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html