LECTURE 13: REGULAR EXPRESSIONS AND TEXT PROCESSING - CONTINUED

Adapted from:
Richard J. Barbalace ,Alex Rolfe, Perl Programming,
sipb-iap-perl@mit.edu

OUTLINE

- Regexes (REs) perform textual pattern matching
- Regexes maybe quoted in several ways
- Regexes are their own mini-language
  - Match letters, numbers, other characters
  - Exclude certain characters from matches
  - Match boundaries between types of characters
  - Group subpatterns
  - Match repetitions
- Perl has several regex operators

GROUPS

- Group subpatterns: () group a subpattern
- (Mon|Tues|Wednes|Thurs|Fri|Satur|Sun)day
- Used not only for matching, but also retrieving sub-parts
- Match repetitions
  - \1 and \2 refer to the first and second matched groups in the pattern
- Example: groups.pl

GROUPS

- ((cat)|dog): There are two capture groups
  - If the input string is dog, capture group 1 is dog, 2 is undefined
- ([abc])\1: find aa, bb, or cc
- (1|2)(3|4)\1\2
  - 1313
  - 2424
  - ....

OPERATORS

- Perl has several regex operators
  - m just matches, returning Boolean
  - s/match/replacement/ substitutes
  - tr/class/replacement/ transliterates
- See operator.pl

MODIFIERS

- Perl has several regex modifiers
  - g is global, allows for multiple substitutions
  - i is case insensitive
  - s treats string as one line
- See modifiers.pl

IN PYTHON?

- import re
- re.match(pattern, sequence)
  - Can be casted to bool, or/and used in a conditional
  - Searches only at the beginning
- re.search(pattern, sequence).group()
  - Searches anywhere
- re.findall(pattern, sequence)
  - Finds all matches
- re.compile(pattern)
  - In case RE is reused
  - re_compiled = re.compile(pattern)
    re.search(sequence)
- re.sub(pattern, repl, sequence)

MORE EXAMPLES

Source: Sam Hughes, Learn regular expressions in about 55 minutes
http://qntm.org/files/re/re.html

LITERALS AND THE DOT

- cat: find a c, followed by a, followed by t
- c.t: find a c, followed by any single character except newline, followed by t
- c\.t: c, followed by dot, followed by t
- c\\t c, followed by backslash, followed by t
- c[.]t or c[\.]t same as c\.t

CHARACTER CLASS EXAMPLES

- Order and repetitions are not important
  - [aabbbccc] is the same as [cba]

- c[aeiou]t: c followed by a vowel followed by t (i.e., cat,cet,cit,cot,cut)

- [0123456789]: find a digit, same as \d or [0-9]

- \[a\] find a left square bracket followed by an a followed by a right square bracket

CHARACTER CLASS RANGE AND NEGATION

- [A-Z]: find an upper-case letter
- A-Z: find an A followed by a hyphen followed by a Z
- [0-9.,]: find a digit or a full stop or a comma
- [0-9a-fA-F]: find a hexadecimal digit
- [a-zA-Z0-9\-]: find an alphanumeric character or a hyphen
- [1-31]: find a 1 or 2 or a 3
- [^a]: find any character other than an a
- [^a-zA-Z0-9]: find a non-alphanumeric character

MULTIPLIERS

- a{1}: find an a
- a{3}: find ”aaa”
- [abc]{2}: find a or b or c, followed by a or b or c
  - aa, ab, ac, ba, bb, bc, ca, cb,cc

ALTERNATION

- Uses the | operator
- cat|dog means cat or dog
- ca[td]og is different than cat|dog
- a|b|c is same with [abc]
- [cat|dog] is different (c, a, t, d, o, g or a pipe)

IP EXAMPLE

- Let's try it

MATCHING ON A LOOP

- See match.pl and match-list.pl

TEXT PROCESSING

- Counting number of words in a file
  - See word-count.pl
- Counting number of ‘the’ words in a file
  - See the-counter.pl
- Count the number of words that has #N letters
  - See varlength.pl
- Count –ly words
  - See ly.pl

FIN!