Regular Expressions

Regular Expressions

- A regular expression is a special text string defining a search pattern for matching text.
- Regular expressions are used in many Unix utilities.
  - like grep, sed, vi, emacs, awk, ...
- The form of a regular expression:
  - It can be plain text ...

  > grep unix file (matches all the appearances of unix in file)
  - It can also be special text ...

  > grep '[uU]nix' file (matches unix and Unix)

Regular Expressions and Filename Expansion

- Regular expressions are different from file name expansion.
  - Regular expressions are interpreted and matched by special utilities (such as grep).
  - File name expansions are interpreted and matched by shells.
  - They have different wildcarding systems.
  - Filename expansion takes place first!

compute[1] > grep '[uU]nix' file

compute[2] > grep [uU]nix file (try yourself)

Regular Expression Wildcards

- A dot . matches any single character

  a.b matches axb, a$b, abb, a.b

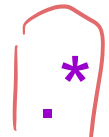
  but does not match ab, axxb, a$bccb

- * matches zero or more occurrences of the previous single character pattern

  a*b matches b, ab, aab, aaab, aaaab, …

  but doesn't match axb

- What does the following match?



任意字符

Regular Expression Wildcards

- + matches one or more occurrences of the previous single character pattern

  a\+b matches ab, aab, aaab, aaaab, …

- ? matches zero or one occurrence of the previous single character pattern

  a\?b matches b and ab

- + and ? have to be escaped with \ to have the special meaning
- Use \(r\) with *, \+, and \? if r is not just a single character

Character Sets and Ranges

- Matching a set or range of characters is done with [...]
  - [wxyz] - match any of w, x, y, z
    [u-z] - match a character in range u - z
- Combine this with * to match repeated sets
  - Example: [aeiou]* - match any number of vowels
- Wildcards lose their specialness inside [...]
  - If the first character inside the [...] is], it loses its specialness as well
  - Example: '[])}]' matches any of those closing brackets

Match Parts of a Line

- Match beginning of line with ^ (caret)

  ^TITLE

  - matches any line containing TITLE at the beginning
  - ^ is only special if it is at the beginning of a regular expression

- Match the end of a line with a $ (dollar sign)

  FINI$

  - matches any line ending in the phrase FINI
  - $ is only special at the end of a regular expression
  - Don't use $ and double quotes (problems with shell)

- What does the following match? ^WHOLE$

Matching Parts of Words

- Regular expressions have a concept of a "word" which is a little different than an English word.
  - A word is a pattern containing only letters, digits, and underscores (_)
- Match beginning of a word with: \<
  - \<Fo matches Fo if it appears at the beginning of a word
- Match the end of a word with: \>
  - ox\> matches ox if it appears at the end of a word
- Whole words can be matched too: \<Fox\>

More Regular Expressions

- Matching the complement of a set by using the ^
  - [^aeiou] - matches any non-vowel
  - ^[^a-z]*$ - matches any line containing no lower case letters
- Regular expression escapes
  - Use the \ (backslash) to "escape" the special meaning of wildcards
    - CA*Net
    - This is a full sentence\.
    - array\[3]
    - C:\\DOS
    - \[.*\]

Regular Expressions Recall

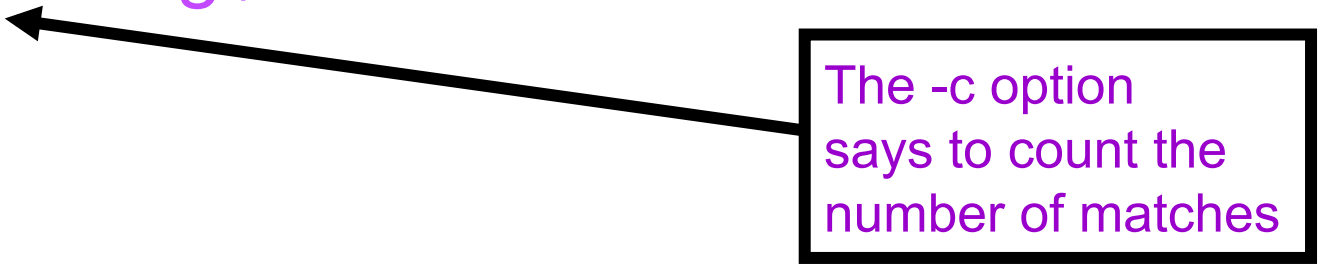
- A way to refer to the most recent match
- To remember portions of regular expressions
  - Surround them with \(...\)
  - Recall the remembered portion with \n where n is 1-9
    - Example: '^\([a-z]\)\1'
      - matches lines beginning with a pair of duplicate (identical) letters
    - Example: '^.*\([a-z][a-z]*\).*\1.*\1'
      - matches lines containing at least three copies of something which consists of lower case letters

Matching Specific Numbers of Repeats

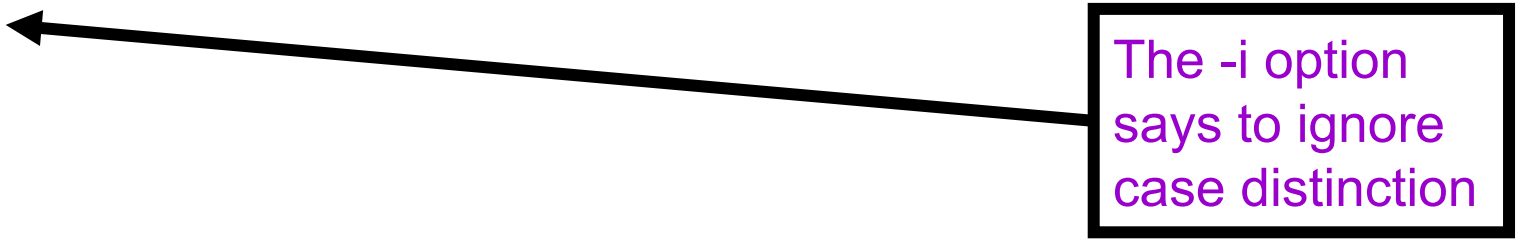
- X\{m,n\} matches m -- n repeats of the one character regular expression X
  - E.g. [a-z]\{2,10\} matches all sequences of 2 to 10 lower case letters
- X\{m\} matches exactly m repeats of the one character regular expression X
  - E.g. #\{23\} matches 23 #s
- X\{m,\} matches at least m repeats of the one character regular expression X
  - E.g. ^[aeiou]\{2,\} matches at least 2 vowels in a row at the beginning of a line
- .\{1,\} matches more than 0 characters

Regular Expression Examples (1)

- How many words in /usr/share/dict/words end in ing?
  - grep -c 'ing$' /usr/share/dict/words



- How many words in /usr/share/dict/words start with un and end with g?
  - grep -c '^un.*g$' /usr/share/dict/words
- How many words in /usr/share/dict/words begin with a vowel?
  - grep -ic '^[aeiou]' /usr/share/dict/words



Regular Expression Examples (2)

- How many words in /usr/share/dict/words have triple letters in them?
  - grep -ic '\(.\)\1\1' /usr/share/dict/words
- How many words in /usr/share/dict/words start and end with the same 3 letters?
  - grep -c '^\(...\).*\1$' /usr/share/dict/words
- How many words in /usr/share/dict/words contain runs of 4 consonants?
  - grep -ic '[^aeiou]\{4\}' /usr/share/dict/words

Regular Expression Examples (3)

- What are the 5 letter palindromes present in /usr/share/dict/words?
  - grep -ic '^\(.\)\(.\).\2\1$' /usr/share/dict/words
- How many words in /usr/share/dict/words have y as their only vowel
  - grep '^[^aAeEiIoOuU]*$' /usr/share/dict/words | grep -ci 'y'
- How many words in /usr/share/dict/words do not start and end with the same 3 letters?
  - grep -ivc '^\(...\).*\1$' /usr/share/dict/words

The -v option says to select non-matching lines

Extended Regular Expressions (1)

- Used by some utilities like egrep or grep –E support an extended set of matching mechanisms.
  - Called extended or full regular expressions.
  - Less use of escape character \, but no recall
- + matches one or more occurrences of the previous single character pattern.
  - a+b matches ab, aab, ... but not b (unlike *)
- ? matches zero or one occurrence(s) of the previous single character pattern.
  - a?b matches b, ab
- + and ? do not need \ to have the special meaning in extended regular expression.

Extended Regular Expressions (2)

- r1|r2 matches regular expression r1 or r2 (| acts like a logical "or" operator).
  - red|blue will match either red or blue
  - Unix|UNIX will match either Unix or UNIX

- (r) allows the *, +, or ? matches to apply to the entire regular expression r, and not just a single character.
  - (ab)+ requires at least one repetition of ab

Extended Regular Expressions (3)

- '\(r1\).*\1' or '(r1).*\1' (recall) is not working for extended regular expression.
  - '[^aeiou]\{4\}' is replaced by '[^aeiou]{4}'
  - '(r1){4}' can be used
- character classes are predefined.
  - [:lower:] for a-z
  - [:upper:] for A-Z
  - [:alpha:] for A-Za-z
  - [:digit:] for 0-9
  - [:alnum:] for A-Za-z0-9