Regular Expressions

craig@ccrma.stanford.edu

29 April 2013

Basic Regular Expressions

^	beginning of line anchor
$	end of line anchor
.	any one character
*	one or more of the preceding atom
[ ]	one of the characters from the set
\	metacharacter escape. For example \$ means a dollar sign, not end-of-line

grep

Generalize Regular exPression

grep word file.txt	Search for lines in file.txt which contain “word”
grep ^word file.txt	search for lines in file.txt which start with “word”
grep ^..b file.txt	search for lines where the 3rd character is “b”.
grep -i ^..b file.txt	equivalent to grep ^..[Bb] file.txt
grep “^..b” file.txt	wise idea to place regular expressions in quotes (single or double) to avoid parsing by shell before sending the search string to grep as an argument.

Cheating at Hangman

E N _ O _ _ E _ E N T

```
grep –i '^en.o..e.ent$' /usr/share/dict/*
```

endorsement
enforcement
engorgement
enlodgement
ennoblement

http://hangman.no → frequently misspelled words

[ ] operator

[123]	match to 1, 2, or 3
[1-3]	match to 1, 2, or 3
	- between two characters means a range (in ASCII) of characters
[-1-3]	match to minus sign, 1, 2, or 3
[1-3-]	match to 1, 2, 3, or minus sign
	- before/after characters in list is not range, but rather minus sign char.
[A-Z]	match to any capital letter
[A-Za-z]	match to any letter
[0-9]	match to any digit
[0-9A-Fa-f]	match to any hex digit
[^AEIOU]	any character other than A, E, I, O, or U
	^ as first character in [] means negate list of characters
[AEIOU^]	match to A, E, I, O, U, or ^ character (^ is a regular character if not first)
[\^AEIOU]	same as above (^ is escaped, so just a regular character)
[]ABC]	match to], A, B, or C (when] is first in list, then just a regular character)
[ABC[]	match to A, B, C, or [(when [is last in list, then just a regular character)

Character Sets

[:alnum:]	alpha-numeric character, equivalent to [0-9A-Za-z] or \w
[:alpha:]	alphabetic character, equivalent to [A-Za-z]
[:cntrl:]	a control character
[:digit:]	numeric character, equivalent to [0-9]
[:graph:]	?
[:lower:]	lower-case character, equivalent to [a-z]
[:upper:]	upper-case character, equivalent to [A-Z]
[:print:]	printable character
[:punct:]	punctuation character
[:space:]	space character
[:xdigit:]	hexadecimal digit [0-9a-fA-F]

Extended Regular Expressions

?	0 or 1 of the preceding atom
+	1 or more of the preceding atom
|	Logical or
( )	Atomic grouping
{ }	Generalized repetition {1,5} = repeated 1 to 5 times

More regular expressions

(23)+	match one or more patterns of “23” in a row: 23 2323 232323
[0-9]+\.[0-9]*	match one or more digit followed by a decimal point followed by 0 or more digits in the fraction.
^.*$	match to any line
^….$	match any line which has four characters in it
^.{50}$	match any line with 50 characters in it
^.{20,30}$	match any line which has 20 to 30 characters in it
(sharp|flat)	will match lines containing either a sharp or a flat

Basic v Extended regex

- grep by default uses basic regex. Add –E option or use “egrep” for extended set.
- Use egrep or “grep –E” for extended set.
- Use extended set in basic mode by escaping character:

grep “A+”	search for the string “A+”
grep –E “A+”	search for one or more A’s in a row
egrep “A+”	search for one or more A’s in a row
grep “A\+”	search for one or more A’s in a row
egrep “A\+”	search for the string “A+”

PERL regular expressions

Further generalizations of regular expressions

\d	= [:digit:] or [0-9]
\s	= [:space:] or [\t\n\r]
\S	= not a space character
\b	= word boundary
\B	= not a word boundary
\w	= word character [a-zA-Z0-9_]
\W	= not a word character

Look ahead/behind (http://www.perlmonks.org/?node_id=518444)

cat(?=s)	match to “cat” if it is followed by “s”
cat(?=[^s])	match to “cat” if it is not followed by “s”
cat(?!s)	match to “cat” if it is not followed by “s”
(?<=s)cat	match to “cat” if it is preceded by “s”
(?<=[^s])cat	match to “cat” if it is not preceded by “s”
(?<!s)cat	match to “cat” if it is not preceded by “s”

grep –P	turns on PERL regular expression syntax
pgrep	shorthand for grep -P

Unix programs dealing with regex

awk	pattern-action language
perl	similar to awk, but newer
ed	line-oriented text editor
vi/ex	full-screen text editor
expr	shell expression evaluator
grep	file searching
sed	stream editor

```
sed 's/cat/dog/g' filein.txt > fileout.txt
```

Change all occurrences of "cat" in filein.txt to "dog" and save the result to fileout.txt.

Humdrum programs w/regex

correl	measure numerical similarity between two spines
fields	list spine/field/structure of a Humdrum file
ditto	replace null tokens with previous data token
hint	harmonic intervals
humsed	stream editor (sed) for Humdrum files
mint	melodic interval
num	number selected records
patt	locate and output user-defined patterns
pattern	exhaustively locate and count user-defined patterns
recode	recode numeric tokens in selected Humdrum spines
regexp	interactive regular-expression tester
rend	split data tokens into subtokens
scramble	randomize order of Humdru data
xdelta	calculate numeric differences between successive data tokens
yank	extract passages from a Humdrum file
ydelta	calculate numeric differences from concurrent data

Searching for sonorities

tntype -a jrp://Jos3010 | **hgrep** 4-29B --mark | myank --marks

```
**kern  **kern  **kern  **kern  **kern  **kern    **tnt
*clefF4 *clefF4 *clefGv2*clefGv2*clefGv2*clefG2   *
*k[]    *k[]    *k[]    *k[]    *k[]    *k[]      *
*M2/1   *M2/1   *M2/1   *M2/1   *M2/1   *M2/1     *
*met(C|)*met(C|)*met(C|)*met(C|)*met(C|)*met(C|)  *
=6      =6      =6      =6      =6      =6        =6
!!LO:TX:Z=20:X=-90:t=6
0r      2E\     0r      2c\]    2e\     2g/       3-11B
.       2F\@    .       4B\@    2c\@    2a/@      4-29B
.       .       .       4A/     .       .         3-11B
.       2G\     .       2B\     1g      2d/       3-11B
.       2C/     .       2c\     .       4g/       2-5
.       .       .       .       .       4a/       3-7A
=       =       =       =       =       =         =
*-      *-      *-      *-      *-      *-        *-
```

Searching for sonorities

```
tntype -a jrp://Jos3010 | hgrep 4-29B --mark | myank --marks | hum2muse | \
    muse2ps =z21c200 | convert –quality 100 –density 300 - -trim –resize 50% output.png
```