



Regular Expressions

CS 2204

Class meeting 6



What is a regular expression (RE)?

- A pattern
- Defines a set of strings of characters
- Any string in the set is said to be “matched” by the RE (the RE matches the string)



Why REs?

- Pattern matching is a useful tool in many real-world situations:
 - Search for a file on a filesystem
 - Find and replace strings in a file
 - Extract particular data elements from a database
- REs are an important part of formal languages - one of the basic CS theory disciplines



UNIX programs that use REs

- `grep/egrep` (search within files)
- `vi/emacs` (text editors)
- `ex` (line editor)
- `sed` (stream editor)
- `awk` (pattern scanning language)
- `perl` (scripting language)



Characters vs. metacharacters

- In patterns, characters can be any character except a newline
- Metacharacters are special characters that have a special meaning
- To use metacharacters as regular characters in a pattern, quote them with the `'\'` character



Using egrep

- `egrep pattern filename(s)`
- To be safe, put quotes around your pattern
- Examples:
 - `egrep "abc" file.txt` (print lines containing "abc")
 - `egrep -i "abc" file.txt` (same, but ignore case)
 - `egrep -v "abc" file.txt` (print lines not containing "abc")
 - `egrep -n "abc" file.txt` (include line numbers)



Metacharacters 1

- Period (.): matches any single character
 - “a.c” matches abc, adc, a&c, a;c, ...
 - “u..x” matches unix, uvax, u3(x, ...
- Asterisk (*) matches zero or more occurrences of the previous RE
 - Not the same as wildcards in the shell!
 - “ab*c” matches ac, abc, abbc, abbbc, ...
 - “.*” matches any string



Metacharacters 2

- Caret (^): means beginning of line
 - “^D” matches all strings starting with D
- Dollar sign (\$) means end of line
 - “d\$” matches all strings ending with d
- Backslash (\): used to quote other metacharacters
 - “file\.txt” matches file.txt but not fileatxt



Metacharacters 3

- Square brackets ([]) indicate a set/range of characters
 - Any characters in the set will match
 - ^ before the set means “not”
 - - between characters indicates a range
 - Examples:
 - “[fF]un” matches fun, Fun
 - “b[aeiou]g” matches bag, beg, big, bog, bug
 - “[A-Z].*” matches any string beginning with a capital letter
 - “[^abc].*” matches any string not starting with a, b, or c



Metacharacters 4

- Plus (+): matches one or more occurrences of the preceding RE
 - “ab+c” matches abc, abbc, but not ac
- Question mark (?): matches zero or one occurrences of the preceding RE
 - “ab?c” matches ac, abc but not abbc
- Logical or (|): matches RE before or RE after bar
 - “abc|def” matches abc or def



Metacharacters 5

- Parentheses (): used to group REs when using other metacharacters
 - “a(bc)*” matches a, abc, abcabc, abcabcabc, ...
 - “(foot|base)ball” matches football, baseball
- Braces ({ }): specify the number of repetitions of an RE
 - “[a-z]{3}” matches three lowercase letters
 - “m.{2,4}” matches strings starting with m between three and five letters



What do these mean?

- `egrep "^B.*s$" file`
- `egrep "[0-9]{3}" file`
- `egrep "num(ber)? [0-9]+" file`
- `egrep "word" file | wc -l`
- `egrep "[A-Z].*\?" file`



Practice

- Construct `egrep` commands that find in `file`:
 - Lines beginning with a word of at least 10 characters
 - Lines containing a student ID number in standard 3-part form
 - Number of lines with 2 consecutive capitalized words
 - Number of lines not ending in an alphabetic character
 - Lines containing a word beginning with a vowel at the end of a sentence