Regular Expression Review

Homework 2

4 Basic Concepts

- Quantification
  - How many times of previous expression?
  - Most common quantifiers: ?, *, +
- Grouping
  - Which subset of previous expression?
  - Grouping operator: ()
- Alternation
  - Which choices?
  - Operators: [] and |
- Anchors
  - Where?
  - Characters: ^ and $

RegEx Exercises

- Which of the following strings would match the regular expression: aab?b
  - A. aabb
  - B. aabbb
  - C. aab

RegEx Exercises

- Which regular expression would match the words “favorite” and “favourite”?
  - Answer: “favou?rite”

RegEx Exercises

- Which regular expression would match the words “Ggle”, “Gogle”, “Google”, Gooogle, etc.?
  - Answer: “Go*gle”

- Which one would match “Gogle”, “Google” and “Gooogle” but not “Ggle”?
  - Answer: “Go+gle”

RegEx Exercises

- Which regular expression would match any version of the word “Google” that has an even number of o’s?
  – Answer: “G(oo)+gle”

- Which regular expression would match any version of the word “Google” that has fewer than 7 O’s?
  – Answer: “Go{0,6}gle”

RegEx Exercises

- Which line(s) would this regular expression match? “^T.+e$”
  - A. The
  - B. Te
  - C. Three
  - D. Then

RegEx Exercises

- Which regular expression(s) would match the words “Ted”, “Ned” and “Sed”?
  - A. (T|N|S)ed
  - B. [TNS]ed
  - C. .ed
  - D. [L-U]?ed
  - E. *ed

RegEx Exercises

- Which regular expression would match all subdirectories within a directory?
  - Answer: ls –l | grep “^d”

Backreferencing

- Refer to a match that was made earlier in the regex
- Ex1: Replace all words in <> in a file, with just words
  - **Answer:** **`cat file.txt | sed 's/<\(.*\)>/\1/g'`**
- Ex2: Find all lines in a file that start and end with the same word
  - **Answer:** **`cat test2 | grep "^\([A-Za-z]\+\).*\1$"`**

`grep` Matches **Lines**!

- echo “To be or not to be” | grep “be$”
  - Output: “To be or not to be”

- echo –e “To be or\nnot to be” | grep “T.*e$”
  - Output: Nothing

  Reasons:

  1. Neither “To be or” nor “not to be” match the regular expression
  2. ‘.’ character often matches any single character except newline character

Homework 2

- Write a script `sameln` that does the following:
  - User provides a directory name as an argument
  - Finds all regular files in directory and ignores all other types (directories, symlinks, etc.)
  - If 2 or more files have the same content (cmp)
    - Keep the file whose name is alphabetically first or starts with a dot
    - Replace duplicates with hard links (ln)

    - File names may contain special characters!

Sample Code

```
#!/bin/bash
        dir=$1
        RESULT=`ls -a $dir`
        declare -a ARRAY                        #”man declare” –a option declares an array
        let count=0
        for FILE in $RESULT
        do
            if [ ! –L "$dir/$FILE" ]
            then
                    if [ -f "$dir/$FILE" ]
                    then
                            echo "$dir/$FILE is a regular file."
                            ARRAY[$count]="$dir/$FILE"
                            let count=count+1
                    else
                            echo "$dir/$FILE is NOT a regular file.”
                    fi
            fi
        done
        echo “There are $count regular files.”
```

Checking Hard Links

- Inode: data structure that stores information about files
  - File type
  - Permission
  - Owner
  - File Size, etc.
- Each inode is identified by a unique inode number within the file system
- Check a file's inode number: ls –i filename
- **How do you check if two files are hard-linked?**
  - Same inode number

Homework Hints

- Iterate over an array in bash
  - for i in "${arrayName[@]}"