MSDL
Modelling, Simulation and Design Lab

Regular Expressions

October 13/16, 2009

Computersystemen en –architectuur

2009 –2010

Regular Expressions

- What?
  - A set of characters specifying a *pattern*
- Why?
  - Search for patterns in *lines* of text
  - Flexible and powerful
- Caveat
  - Different from file matching patterns!
  - Different standards... (Solaris, POSIX, Emacs, Perl, ...)
- Extended regular expressions

Anchors

- ^ (beginning of line) and $ (end of line)
- Examples
  - ^A "A" at the beginning of a line
  - A$ "A" at the end of a line
  - A^ "A^" anywhere on a line
  - $A "$A" anywhere on a line
  - ^^ "^" at the beginning of a line
  - $$ "$" at the end of a line

Character Sets

- Simplest character set:
  - `abc` matches the character sequence abc
- Want to use literal characters that have a special meaning?
  - "Escape" with backslash "\"
- Special characters
  - `.` any single character
- Ranges
  - Between "[" and "]": one of these characters/patterns
    - "[^" and "]": NOT one of these characters/patterns
  - Use "-" between characters to denote a range between these characters
    - See ASCII table

Character Sets

- Examples of ranges
  - `[]` The characters "[]“
  - `[0]` The character "0"
  - `[0-9]` Any number
  - `[^0-9]` Any character other than a number
  - `[-0-9]` Any number or a "-"
  - `[0-9-]` Any number or a "-"
  - `[^-0-9]` Any character except a number or a "-"
  - `[]0-9]` Any number or a "]"
  - `[0-9]]` Any number followed by a "]"
  - `[0-9-z]` Any number, or any character between "9" and "z".

Modifiers

- Repeating character sets: use ”*”, “+”, “?”
  - Repeat preceding character set zero or more times
- A specific number of repetitions: use “{“ and “}”
  - Use two numbers, seperated with a comma, for bottom an upper limit of repetitions
  - Ommit last number to allow unlimited repetitions
  - Use only one number for a precise number of repetitions
- Choice: use “|”
- Match only words: use “\<“ and “\>”
  - Surrounding characters are anything but a letter, number, underscore, new line or end of line

Modifiers

*	Any line
*	Any line with an asterisk
\\	Any line with a backslash
^*	Any line
^A*	Any line
^A*	Any line starting with an "A*"
^AA*	Any line if it starts with one "A"
^AA*B	Any line with one or more "A"'s followed by a "B"
^A{4,8}B	Any line starting with 4, 5, 6, 7 or 8 "A"'s followed by a "B"
^A{4,}B	Any line starting with 4 or more "A"'s followed by a "B"
^A{4}B	Any line starting with "AAAAB"
A\{4,8\}	Any line with "A{4,8}“
USA|USSR	Any line containing “USA” or “USSR”
\<[Tt]he\>	Any line containing the word “the” or “The”

Backreferences

- Reuse patterns
  - Surround pattern with “(“ and “)”
  - Use \1, \2, \3, ... to reuse previously matched patterns
  - Examples
    - `([a-z])\1` Any line with two consecutive characters
- Also used for grouping
  - `(bla){2}` Any line containing “blabla”

Variants

Notation	awk	ed	egrep	expr	gres	pg	sed	vi
.	•	•	•	•	•	•	•	•
^	•	•	•		•	•	•	•
$	•	•	•	•	•	•	•	•
[…]	•	•	•	•	•	•	•	•
[::]	•	•	•	•	•	•	•	
*re******	•	•	•	•	•	•	•	•
*re***+**	•		•		•	•		
*re***?**	•		•		•	•		
*re***|***re*	•		•		•	•		
*****d	•	•	•	•	•	•	•	•
(…)	•		•		•	•		
\(…\)		•		•			•	•
\<			•					•
\>			•					•
\{ \}	•		•		•	•	•	
{ }	•		•					

Sed

- Stream Editor
  - gsed –r for extended regular expressions
- Command s for substitution
  - Syntax:

    `gsed -r 's/from/to/'`

    match expression and replace expression on a line

    `gsed -r 's/from/to/g'`

    replace all occurences on each line (global replacement)
- Usage:
  - `echo "hello world" | gsed -r 's/from/to/'`
  - `gsed -r 's/from/to/' < file.txt`
  - `gsed -r 's/from/to/' < file.txt > file2.txt`

Sed

- Sed substitution algorithm
  1. Read the stream character by character
  2. When matching, be as greedy as possible
  3. Replace matched pattern
  4. (Continue after replacement - /g global replacement)

Sed

- Use '/' character in your expression?
  - "Escape" with '\'

```
gsed -r 's/\/usr\/bin\/bash/ bash shell/' < /etc/passwd
```

  - Use another delimiter

```
gsed -r 's:/usr/bin/bash: bash shell:' < /etc/passwd
```

- Reuse matched string in replace string
  - Use '&'

```
gsed -r 's:/usr/bin/bash: shell = &:' < /etc/passwd
```

    - equals:

```
gsed -r 's:(/usr/bin/bash): shell = \1:' < /etc/passwd
```

Sed options

- -e combine commands
- -f read commands from script file
- -n silent mode

Sed commands

- Specify line numbers
  - `gsed -r '1,10 s/A/B/g'`
- Combine command flags when it makes sense!
  - g global replacement
  - w file write to a file
  - c line replace matched line with given line
  - i line insert line after match
  - a line append line after match
  - r file append file after match
  - p force to print the modified lines
  - d deletes matched line