ICS215: Assignment 3

Due on Wednesday, October 9, 2013 at the beginning of the lecture

Dr. Jan Stelovsky
Halawai Online Lecture MW 10:30am

Please hand in scripts by email to Andreas Brauchli andreasb+ics215@hawaii.edu

Andreas Brauchli andreasb+ics215@hawaii.edu

September 26, 2013

Regular expression assignment with Perl

This assignment should build up your regex skills. To refine those, we will continue to use Perl's regular expression engine.

All problems must be completed by the hand-in date. It is however strongly suggested to start early, as you may need the time to complete the tasks.

Problem 1

Multiple choice - min 0, max 6pts
Are the following answers correct or incorrect? Every correct answer +1, every wrong answer -0.5pts, unanswered 0pts

1. The regular expression and its flags `s/foo/bar/` applied..
   A. on input "Foo" produces "bar"
   B. on input "bar" produces "bar"
   C. on input "foofoobar" produces "barbarbar"
2. The regular expression and its flags `s/my\s(\w{3})\sis\s([rR]ed)/rat\1/` applied..
   A. on input "my cat is red" produces "my rat is red"
   B. on input "my dog is red" produces "my rat is red"
   C. on input "my frog is red" produces "my frog is red"

Problem 2

Phone numbers - 9pts
Create one regular expression that validates US phone numbers and transforms them into this format: (808) 123 4567. The phone number inputs may be in following formats:

- 8081234567
- 808-123-4567
- 808 123 4567
- (808) 123 45 67
- +18081234567
- +1 808 123 4567

You can accept other formats if the number remains valid but you must be explicit about what other formats you accept.

Problem 3

Parsing Perl - 10pts
Create regular expressions that parses the following subset of perl instructions. These are your constraints:

- Whitespaces can be any number of tabs or spaces or a combination thereof.
- Variable and function names may contain lower and upper case characters, numbers and the underscore (_) but must start with a letter and may not be empty.
- Arrays must be accessed by a constant or scalar variable that is not an other array's element. `$array[$i]` is possible but not `$array[$subarray[$i]]`.
- Quoted strings may contain all valid characters but are guaranteed not to contain any quotation marks, not even when escaped (this will make it easier.) The two allowed quotation marks are double-quotes (") and single-quotes ('). The quoted string (inside quotes) may also be empty.

State one regular expression for each item.
You may use placeholders when reusing a previous regex:
BINARY = /0|1/;
DECIMAL = /BINARY|[2-9]/.
This is strongly suggested to break down complexity and avoid mistakes.

- Scalar and array variables declaration with the `my` keyword: my $i; | my @a;
- Quoted strings not containing the quotation mark: "hello Perl" | 'regex string matching is fun'
- Subroutine calls with semicolon: foo(); | foo (3, "hi"); | chomp; | chomp();
- Comparison expressions: $i == 3 | 4 < 3 | $i >= $a[2]
- Assignments: $i = 3; | $a[3] = 3; | $myarray[$i] = "foo";