CS202: PROGRAMMING PARADIGMS & PRAGMATICS

Semester II, 2021 – 2022

Lab 6: Exercises in Regular Expressions

- **Aim:** The goal of this lab is to get hands-on experience with using Regular Expressions.

- **Let's get started!**

  a. Create a directory structure to hold your work for this course and all the subsequent labs:
     - Suggestion: `CS202/Lab6`

  b. Write scripts / code to implement regular expressions for the following exercises in Perl!

  c. For exercise 1 and 2 below, the program should take a string as an input and display either "**ACCEPTED**" or "**REJECTED**"

- **Exercises**

  - You are in the market to buy a red pick-up truck, and you wish to develop an automated web searching program (a spider) to search daily through various online newsgroups and classified ad websites to find text containing the word red and the phrase pick-up truck close to each other, followed by a price. Specifically, you should match the words red and the phrase (pickup/pick-up/pick up) truck separated by at most two other words in between. The pick-up truck phrase could appear before or after the word red. After the words red and the phrase pick-up truck, the text should also contain a price. Sample text strings that should be accepted / rejected by the RE are given below: (`Truck.pl`)

ACCEPT	REJECT
• red pickup truck $5000 • red pickup truck $5,000 • red pickup truck $1,234.56 • red pick-up truck $5000 • red pick up truck $5000 • red toyota pick-up truck $5000 • red toyota 1993 pick-up truck $5000 • blah blah red toyota 1993 pick-up truck blah blah $5000 blah • pickup truck red $5000 • pick-up truck 1993 toyota red $5000 • blah blah blah pick-up truck toyota 1993 red blah blah blah $5000 • desperate: red 1993 toyota pickup truck for sale. $2,000 o.b.o. • toy pickup truck - cherry red: $12. • red red pickup pickup truck truck $5000.	• Red • Truck • pickup truck • red pickup truck • red $5000 • pickup truck $5000 • red truck $5000 • $5000 red pickup truck • blue pickup truck $5000 • red car $5000 • red toyota 1993 pick-up truck • red 1993 toyota automatic pick-up truck $5000 • fred's pick-up truck sold for $5000 • pick-up trucks by fred: $5000 • reddy for sale pickup truck: $5000)

  - DNA sequences are comprised of a simple 4-alphabet language with the symbols {A,C,G,T}. Three consecutive letters are known as a codon, so ACT and TCG are both codons. A Gene is a collection of at least three codons that starts with an ATG codon and ends with aTAA, TAG, or TGA codon. You need to develop a regular expression that will match strings that contain a gene. Sample DNA sequences that should be accepted/rejected as Genes are given below: (`Gene.pl`)

ACCEPT	REJECT
• ATGCCCTAA • ATGCCCTAG • ATGCCCTGA • CATGCCCTAA • CATGCCCTAG • CATGCCCTGA • CATGCCCTAAC • CATGCCCTAGC • CATGCCCTGAT • TCATGCCCTGACC • TTATGCCCGGGTGACC • AAACTCATGCCCGGGCCCTGACCTTAA • ATGATGATGTAA • ATGAAAAACAAGAATTAA • ATGACAACCACGACTTAA • ATGAGAAGCAGGAGTTAA • ATGATAATCATGATTTAA • ATGCAACACCAGCATTAA • ATGCCACCCCCGCCTTAA • ATGCGACGCCGGCGTTAA • ATGCTACTCCTGCTTTAA • ATGGAAGACGAGGATTAA • ATGGCAGCCGCGGCTTAA • ATGGGAGGCGGGGGTTAA • ATGGTAGTCGTGGTTTAA • ATGTACTATTCATCCTCGTCTTGCTGGTGTTTATTCTTGTTTTAA	• GATTACA • ATGTAA • ATGTAG • ATGTGA • ATGCCCCTAG • ATGCCCCCTAG • CCCATGCCCCTAGCCC • CCCATGCCCCCTAGCCC

- Tokenization is the task of extracting tokens from the input text. The definition of 'token' depends on the application, but in most cases complete words count as tokens; sometimes punctuation markers do as well. Write a simple tokenizer that given an input text and delimiting characters outputs one word per line by replacing strings of delimiting characters with newlines. (`Token.pl`)

- **Submitting your work:**
  - All source files and class files as one tar-gzipped archive.
    - When unzipped, it should create a directory with your ID. Example: **2008CSB1001** (NO OTHER FORMAT IS ACCEPTABLE!!! Case sensitive!!!)
    - Should include: `Truck.pl, Gene.pl, Token.pl, and` README file
    - ***Negative marks for any problems/errors in running your programs***
  - *If any aspects of the tasks are confusing, make an assumption and state it clearly in your* **README**
  - **README** file should also have instructions on how to use/run your program!
  - Submit/Upload it to Google Classroom
    - Marks Allocation: `Truck` [5 points], `Gene` [5 points], `Token` [3 points], README [2 points]