DALHOUSIE UNIVERSITY
Inspiring Minds

Winter 2019 (Jan7-Apr8)
Faculty of Computer Science

CSCI 4152/6509 — Natural Language Processing

Assignment 1

Due: *Friday, Jan 25, 2019 by midnight*
Worth: 67 marks (= 10 + 22 + 10 + 15 + 10)
Instructor: Vlado Keselj, CS bldg 432, 902.494.2893, vlado@dnlp.ca

---

Assignment Instructions:

The submission process for Assignment 1 and later assignments will be based on the SVN system. We assume here that you finished Lab 1, so you are already familiar with SVN (i.e., the Subversion version control system).

SVN instructions: To submit this assignment, you should start with having a new working copy of your SVN course repository in your course directory. It should be a different copy than the copies you used in the Lab 1 (the SVN tutorial lab). Provided that you finished Lab 1 properly, you could even delete the directories 'primary' and 'secondary' from the lab, and their parent directory 'lab1' with their contents. If you are not sure about this, you can also keep it there.

To submit this assignment, Lab 2, and other work in the course, you need to create a new working copy of your SVN course repository that you will use during the term. This copy should have been created in Lab 2, but in case you did not create it yet, here are instructions about how to do it:

Creating SVN working copy if not done in Lab 2: You can login to bluenose and we assume here that you have already created the directory '`csci6509`' or '`csci4152`' in your home directory, depending on your course number. You should go into that directory using the command '`cd csci6509`' or '`cd csci4152`', depending on your course number. You can create a fresh, checked-out copy of your SVN course repository using the following command (only if not done already):

```
svn co https://svn.cs.dal.ca/nlp-course/CSID
```

where CSID is replaced with your actual CS userid, a.k.a., CSID.

Creating a1 directory in SVN: Some questions in the Assignment 1 will require that you finish some labs, and some questions will require a new directory '`a1`', which you will create in the following way: First, you must be in your main directory of your SVN working copy:

`~/csci4152/CSID` or `~/csci6509/CSID`

where '`~`' is your home directory, which may look like `/users/cs/CSID`, `/users/grad/CSID`, or similar. (If you are not familiar with the Unix file system, you should review this as soon

as possible, starting with the commands `cd`, `pwd`, and `mkdir`.) In this directory, you can now create subdirectory 'a1' and add it to SVN using the commands:

```
svn mkdir a1
svn commit -m'a1 created'
cd a1
```

All files with answers must be added to SVN. Do not forget to commit them before the deadline. You can commit the files many times, since the newest files before the deadline will be used.

To clarify the terminology: when we say *submit*, it means that the file must be added to SVN (command `svn add`), and committed to the repository (command `svn commit`).

It is **very important** that you submit files exactly as specified, because the programs that copy the files for the markers depend on this. If you misplace or misname a file or directory, the marker will not receive them and **you will receive 0 marks** for the related question.

1) (10 marks) Complete the SVN lab tutorial as instructed in the Lab 1 and the lab notes.

For this question, you do not need to submit anything. Your work will be directly checked using the directory '`svnlab`' used for the lab.

2) (22 marks, files in *csuserid*/`lab2`) Complete the Lab 2 as instructed. In particular, you will need to properly:

a) (4 marks) Submit the file 'hello.pl' as instructed.

a) (4 marks) Submit the file 'example2.pl' as instructed.

b) (4 marks) Submit the file 'example5.pl' as instructed.

c) (5 marks) Submit the file 'task1.pl' as instructed.

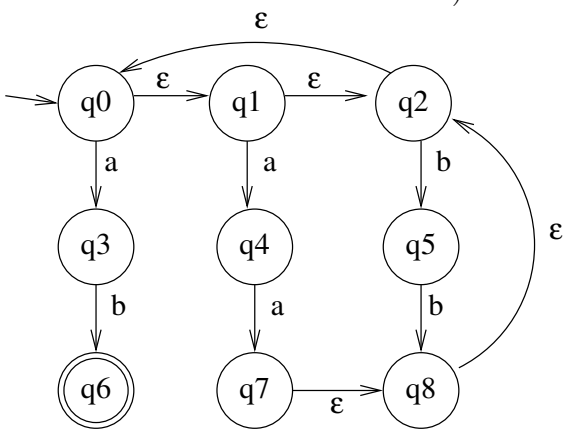
d) (5 marks) Submit the file 'task2.pl' as instructed.

Notice that the examples from (a) and (b) need to compile; if a syntax error got introduced to an example program by your typing mistake or by introducing incorrect characters through copying and pasting from a pdf file, so that the example program does not compile, it will not be accepted. The lab instructions state that the programs should be tested before being submitted.

3) (10 marks) List the levels of NLP and briefly describe all of them in your own words (1–3 sentences for each). Submit your answer as a plain-text file called '`a1q3.txt`'.

4) (15 marks) Submit answer to this question in a plain-text file named `a1q4.txt` submited via SVN in the pathname *CSID*`/a1/a1q4.txt`, where *CSID* is your main SVN course directory.

Consider an NFA (Non-deterministic Finite Automaton) described by the following graph:



a) (5 marks) Give three examples of words accepted by this NFA. Briefly describe what language is accepted by this NFA.

b) (10 marks) Translate this NFA into a DFA using the process discussed in class. Submit your solution as a plain text file named `a1q4.txt` where the DFA is shown as textual table in a format shown below:

```
State      |   a    |    b   |
-----------+--------+--------+
S: q0q1    |  q0q1  |  q0q1  |
-----------+--------+--------+
   q0q1    |  q0q2  |  q0q2  |
-----------+--------+--------+
F: q0q2    |  q0q1  |  q0q1  |
-----------+--------+--------+
```

Exaplanation: Use characters minus, vertical line, and plus to draw the table. The columns correspond to input characters. The DFA states are set of NFA states shows as sequences of states in a sorted order by index (for example, use `q0q1` rather than `q1q0`). use labels `S:` and `F:` to denote start and finish states. If and NFA state is empty set, then use the word '`empty`' to denote it.

5) (10 marks) Submit answer to this question in a file named `a1q5.pl` submited via SVN in the pathname *CSID*`/a1/a1q5.pl`, where *CSID* is your main SVN course directory.

Write a Perl program which does the following:
1) Prints the message: `Enter a positive integer:` with one space after ':' and no newline

character.
2) Reads a number from the keyboard. You can assume that the input is always a positive integer.
3) Prints a square pattern of 'X' characters oof width n, where n is an input number. For example, for $n = 5$ it should print:

```
XXXXX
X   X
X   X
X   X
XXXXX
```

The inside of the 'square' is printed with space characters.