CECS 530 Design of Compilers - Project 1 (100 points)
Posted on Jan 15 (Wed), 2025

<u>Due</u>: Submit your project report to the Blackboard before Feb. 3 (Monday) midnight.

Report Guidelines (No-compliance or late report will be penalized)

[1] Submit your project report in the pdf document format.

[2] Begin your report with a "title page" (including project number, your name, and the date submitted of your report).

[3] Name your report like this <your-last-name>_<initial>_hw1.pdf

[4] All work should be your own writing with clear references of your external sources.

[5] Do not share your report with other students. Any identical or nearly similar reports will get no credit.

[6] If you have any output or results (e.g., screenshots) that need to be submitted then these must be imbedded in your report.

<u>Readings PowerPoints</u>:

1_Introduction.pdf,
2_CaseStudy.pdf, and
2_CFG.pdf

<u>VS2019 Projects</u>:

TreeVizWF

<u>Assignments</u>

1 Install `GraphViz` on your computer and make sure you can run the dot command from
  the command console. For example, on Windows, you need to add the `GraphViz` bin's path to
  the `PATH` environmental variable.

(1.1)(10 points) Use a text editor to write a dot file called, `bst.txt`, to layout a binary search tree
     (BST) consisting of nodes with values 1, 2, 3 …, and 10 of your design (see
     `1_Introduction` or `2_CaseStudy.pdf` for dot BST examples). To draw the BST, use the
     following command (choosing any output file format):

```
$ dot -Tjpg bst.txt -o bst.jpg
```

     List your dot file and show the tree layout in your project report.

(1.2)(10 points) Write a dot file and use the dot command to draw the graph of Seven Bridges of
     Königsberg. In the history of mathematics, Euler's solution of the Königsberg bridge
     problem is considered to be the first theorem of graph theory. Make the drawing
     looking like the one given below:
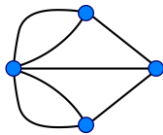
     List your dot file and show the graph layout in your project report.

(1.3)(10 points) The following syntax tree due to Noam Chomsky (1956) is the first syntax tree
     ever appeared in the literature.
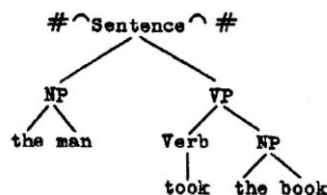
     Write a dot file to generate the above tree (directed or un-directed) using the dot command.
     List your dot file and show the generated tree in your project report.

2 This assignment is to use VS 2019 project, `TreeVizWF`, and TreeBank parenthesized (or bracketed) expressions to visualize the syntax of English sentences.

(2.1) (12 points) The following is quoted from: https://web.stanford.edu/~jurafsky/slp3/18.pdf
(18 Context-Free Grammars and Constituency Parsing), section 18.3 Treebanks:

"A corpus in which every sentence is annotated with a parse tree is called a treebank. Treebanks play an important role in parsing as well as in linguistic investigations of syntactic phenomena.
Treebanks are generally made by running a parser over each sentence and then having the resulting parse hand-corrected by human linguists. Figure 18.5 shows Penn Treebank sentences from the Penn Treebank project, which includes various treebanks in English, Arabic, and Chinese. The Penn Treebank part-of-speech tagset was defined in Chapter 17, but we'll see minor formatting differences across treebanks. The use of LISP-style parenthesized notation for trees is extremely common and resembles the bracketed notation we saw earlier in (18.1). For those who are not familiar with it we show a standard node-and-line tree representation in Fig. 18.6."

The parenthesized (or bracketed) expressions are used in TreeBank to denote a syntax tree. For example, the syntax tree of the sentence,

`I prefer a morning flight`,

can be denoted by this parenthesized expression:
(S (NP (Pro I)) (VP (V prefer) (NP (Det a) (Nom (N morning) (Nom (N flight))))))

The internal node symbols have the following meaning in English grammar:

Symbol	Meaning	Example
S	sentence	*the man walked*
NP	noun phrase	*a dog*
VP	verb phrase	*saw a park*
PP	prepositional phrase	*with a telescope*
Det	determiner	*the*
N	noun	*Dog*
V	Verb	*walked*
P	Preposition	*in*

Use `TreeVizWF` to input the above expression and generate the syntax tree. Show a screenshot of the `TreeVizWF` window application with this input and the generated syntax tree.

(2.2) (8 points) Use the parenthesized expression to define the syntax tree given in problem (1.3) and use `TreeVizWF` to draw the syntax tree. List the parenthesized expression of the tree and show the tree layout in your project report.

3 (10 points) The parenthesized expressions can be used to denote any (rooted) trees as shown in the following examples:

(10 6 (20 15 30))	(60 (16 (9 1 12) (25 19 _)) (80 (64 _ 67) 86))

Use `TreeVizWF` to draw the BST you define in problem (1.1) using dot. Show the parenthesized expression and its generated BST tree.

4 This problem demonstrates the English language is ambiguous (i.e., an English sentence may have multiple syntax trees and as a result, multiple meanings.)

(4.1) (12 points) There are two parenthesized expression files `pocket1.txt` and `pocket2.txt` attached to this project assignment on the Blackboard. These two files contain expressions to generate two different syntax trees for this sentence

John bought a shirt with pockets.

Run `TreeVizWF` with input of the parenthesized expressions given in these two files. Show the generated syntax trees side-by-side (say, in a table like that shown in problem 3).

(4.2) (7 points) According to the syntax trees in (4.1), explain the meanings of the sentence based on the syntax structure. Which meaning of the sentence makes more sense?

(4.3) (6 points) Modify the files, `pocket1.txt` and `pocket2.txt` for this sentence

John bought a shirt with cash.

Run `TreeVizWF` with input of the parenthesized expressions given in these two modified files. In this case, which syntax tree makes more sense semantically?

5 (https://stanfordnlp.github.io/stanza/) Stanza is a collection of accurate and efficient tools for the linguistic analysis of many human languages. Starting from raw text, Stanza divides it into sentences and words, and then can recognize parts of speech and entities, do syntactic analysis, and more. Stanza brings state-of-the-art NLP models to languages of your choosing.

(5.1) (10 points) Use the web app, stanza.run, to find Constituency Parse of the two sentences given in (4.1) and (4.3), respectively. Show screenshots of the results and compare them with those in problem 4.

(5.2) (5 Points) One of the outputs of (5.1) is POS. Explain what POS means.