CS 335 Semester 2022–2023-II: Assignment 2

27^{th} January 2023

Due Your assignment is due by Feb 12 2023 11:59 PM IST.

General Policies

- You should do this assignment ALONE.
- Do not plagiarize or turn in solutions from other sources. You will be PENALIZED if caught.
- We MAY check your submission(s) with plagiarism checkers.

Submission

- Submission will be through Canvas.
- Upload a zip file named "⟨`roll`⟩`-assign2.zip`". The zipped file should contain a directory ⟨`roll`⟩`-assign2`, a sub-directory `problem4`, and the following files:
- Create a zip file named "`cs335_`⟨`roll`⟩`.zip`". The zipped file should contain a folder `assign2` with the following files:
  - A `PDF` file "⟨`roll`⟩`-assign2.pdf`" that contains the solutions for the pen-paper problems, and other details for Problem 4. For Problem 4, describe the tools you used, and INCLUDE compilation and execution instructions.
  - Implementation files in your chosen implementation language.
- We encourage you to use the LaTeX typesetting system for generating the PDF file.
- You will get up to TWO LATE days to submit your assignment, with a 25% penalty for each day.

Evaluation

- Write your code such that the EXACT output format (if any) is respected.
- We will evaluate your implementations on a Unix-like system, for example, a recent Debian-based distribution.
- We will evaluate the implementations with our OWN inputs and test cases (where applicable), so remember to test thoroughly.
- We may deduct marks if you disregard the listed rules.

Problem 1 **[50 marks]**

For the following grammar, design a predictive parser and show the predictive parsing table. Perform desired processing like removing left-recursion and left-factoring on the grammar if required.

$$S \to (L) \mid a$$
$$L \to L, S \mid LS \mid b$$

Problem 2 **[50 marks]**

Show that the following grammar is LALR(1) but not SLR(1).

$$S \to Lp \mid qLr \mid sr \mid qsp$$
$$L \to s$$

Problem 3 **[50 marks]**

Construct an SLR parsing table for the following grammar. Show the canonical set of states and the transition diagram.

$$R \to R \,'|'\, R$$
$$R \to RR$$
$$R \to R^*$$
$$R \to (R)$$
$$R \to a \mid b$$

Note that the vertical bar in the first production is the "or" symbol (i.e., terminal), and is not a separator between alternations.

Resolve the parsing action conflicts in such a way that regular expressions will be parsed normally. Include your disambiguation rules in the `PDF` file, and show the final parsing table.

Problem 4 **[50 marks]**

A dissertation consists of a title and one or more chapters. Each chapter consists of zero or more sections. A section consists of one or more paragraphs. Each paragraph can consist of one or more sentences. Sentences can be of three types: declarative, exclamatory, and interrogative. Declarative, exclamatory, and an interrogative sentence ends with '.', '!', and '?' respectively.

Punctuation marks are period, comma, exclamation, colon, semicolon, and question mark. Sentence separators are only period, exclamation mark, and question mark. Whitespace (not including newline, and can be repeated finite number of times), comma, and semicolon are valid separators between any two words within a sentence. Words cannot be separated by two commas or semicolons.

- You can assume that "Title", "Chapter", and "Section" are keywords, and will not appear in the text body. Furthermore, these keywords are always followed by a ':'.
- A paragraph may start immediately after a Chapter or Section. Two paragraphs are separated by at least one blank line (i.e., two "\n"s).

- A Chapter has to contain one or more paragraphs.
- Two sentences can be separated by one or more whitespace characters, but will not contain a newline character.
- You can assume the words will be well-formed from the English alphabet (e.g., "aPPles" is valid).
- The text may include decimal numbers. Note that integers and floating point numbers (only using decimal point, no scientific notation) may appear in a sentence in a paragraph. You can ignore the leading '+' and '-' characters.
- In case of an invalid input, you can print the first error and exit. Report the line number, additional error information is optional.

We have provided a sample dissertation to elaborate on the specifications. Write a parser for such a semi-structured dissertation. Compute the following statistics.

1. Print the title of the dissertation.
2. Print the total number of chapters, sections, paragraphs, sentences, and total number of words across all paragraphs. That is, ignore the title, chapter, and section lines while counting words. Also, ignore digits and numbers.
3. Print the number of declarative, exclamatory, and interrogative sentences.
4. Pretty print a Table of Contents (ToC) for the dissertation. You should limit till Sections in the hierarchy.

Print in the following format:

```
Title: An Elementary Introduction  to Compiler Design
Number of Chapters: 7
Number of Sections: 9
...
Table of Contents:
Chapter 1: Introduction
    Section 1.1: Contribution One
    Section 1.2: Contribution Two
    Section 1.3: Contribution Three
Chapter 2: Background
Chapter 3: Contribution One
    Section 3.1: Introduction
    ...
```

You are free to use any parser generator (e.g., Bison or ANTLR) and programming language for your implementation. You can use Flex for tokenization if you use Bison. Remember to include the source files and instructions in your submission.