COP701: Assignment #1

July 24, 2024

1 LaTeX to Markdown converter

1.1 Problem Statement

In this assignment, your main objective is to convert LaTeX to an equivalent Markdown document. In pursuance of this objective, you will have to write a LaTeX to Markdown parser from scratch.

The features of LaTeX which you all need to consider are:

- Sections and subsections
- Italics and bold
- Horizontal line (\hrule)
- Paragraph (\par)
- Code blocks (\verbatim)
- Hyperlink
- Images
- Ordered and unordered Lists
- Tables

You can consider other features for extra credit. A sample input LaTeX file and the corresponding output Markdown file is posted on Piazza.

1.2 Workflow and subtasks

The entire assignment can be divided into the following sub-tasks:

- Learn about Markdown and LaTeX.
- Write a lexer i.e. to do a lexical analysis of your LaTeX code and generate a string of tokens. Programs that you can use: flex

- Do not use any available libraries to parse the LaTeX.
- Parse the sequence of tokens using parsers such as yacc and bison (C/C++)
- Generate an AST(Abstract Syntax Tree) of LaTeX code. link
- Map it to an equivalent AST of Markdown.
- Generate the equivalent Markdown document.

1.3 Links to important resources

- Know about the Flex tool and performing lexical analysis using that.
- Some more resources regarding Flex.
- Introduction to an Abstract Syntax Tree.
- Building an Abstract Syntax Tree.
- Bison parser generator.
- YACC parser generator.

2 Logistics

- You have to code in the **C/C++** programming language.
- The **deadline** for this assignment is **25/08/2024 at 11:59 PM**. It is a hard deadline and will not be extended.
- This is an individual assignment (30 Marks)
- You need to create a private git repository either on https://git.iitd.ac.in or github. Git commit history will be checked during evaluation.
- References for coding style can be found on the course webpage. Eg: C/C++.
- You need to write unit tests for your code.
- All the modules of your code need to be documented. Eg: Godoc.
- Use a Makefile for the project build by defining targets, dependencies, and build commands.
- You are not expected to use any external libraries to convert directly to Markdown.
- ANY form of **plagiarism** will not be tolerated.

- Also, create a **run.sh** file, where the first argument will be the name of the LaTeX file and the second argument will be the name of the output Markdown file. We will run the command *./run.sh input.tex output.md* during the evaluation/demo.
- The submission will be made on Moodle. You need to submit all your code (parser, translator) and a report in the PDF format. Compress all these in a tar file with the name $< entry\_number >$ *.tar* and upload on Moodle. The entry number should all be in small letters. For example: 2023mcs2475.tar
- You will be graded on the basis of the output of your code, the coding style, regularity of GitHub commits and your viva/presentation.
- Marks distribution: Coding style - 20%, Git/Documentation - 5%, Demo - 75%
- We will be testing using hidden test cases during the demos.
- Any doubts regarding the course/assignment should be asked on Piazza.