CS39003 Compilers Laboratory, Autumn 2024–2025
Assignment No: 1
Date: 05-Aug-2024

Using LEX on LaTeX files

LaTeX is a typesetting utility written by Leslie Lamport. It is basically a macro package designed on top of Donald Knuth's TeX typesetting system. In this assignment, you are required to write a lex program and a C/C++ program to identify some patterns (commands, environments, math formulas, and comments) in LaTeX files. The input to your program will be a LaTeX file, and the output will be the printing of some statistics (explained near the end) associated with these patterns in the input file.

A TeX or LaTeX file is a text file that stores the instructions on the text, the mathematical formulas, the tables, the figures, and so on, that you want to typeset. Standard text is written as it is. Special formatting requires using commands (also called macros). LaTeX additionally provides some environments for handling a set of specialized items in the document.

The LaTeX patterns relevant to this assignment are explained in detail now.

LaTeX commands There are three types of commands based on the characters used to define them.

- First, there is an active character ~ which, in itself, is a command (its default meaning is a single space where a line cannot be broken).

  Other commands begin with the special character \(backward slash).
- If the command name consists only of alphabetic letters, then the \ will be followed by one of more of the letters a–z and A–Z. Some examples are \a, \abcd, \aone, \aiii. LaTeX commands are case-sensitive. For example, \abc, \Abc, and \ABC are considered different commands. Notice that such command names must contain alphabetic letters only. For example, in \abc12 or \abc_def, we have the command \abc. What follows is not part of the command name.
- Any character other than the alphabetic letters or ~ followed by \ gives a single-character command. For example, you can have commands like _, \1, \\, \$, and so on, whereas in \"u or \"", the command is \".

LaTeX environments A LaTeX environment begins with the special command \begin{env_name} and ends with \end{env_name}. For example, for typesetting right-justified text, one may use the flushright environment as below.

```
\begin{flushright}
Write your right-justified text here.
\end{flushright}
```

You can have spaces between \begin or \end, and the brace-delimited environment name, but spaces inside the braces are not permitted.

Inline math formulas An inline mathematical formula is written between a pair of dollars. For example, the Van der Waals equation $\left(P+\frac{\alpha}{V^2}\right)(V-\beta)=RT$ can be typeset as follows.

```
$\left(P+\frac{\alpha}{V^2}\right)(V-\beta)=RT$
```

Note that the character $ toggles the math mode. Inside the math mode, there may be commands (like \left or \frac). You need to locate these commands as well. Note also that a math-mode formula may span across multiple lines, so you must not search for the terminating $ in the same line as the beginning $.

Displayed math formulas If you want to show your mathematical formulas in separate lines, you can use a pair of $$ in the toggle mode, or the matching commands \[and \]. For example,

```
$$\left(P+\frac{\alpha}{V^2}\right)(V-\beta)=RT$$
```

produces

$$\left(P+\frac{\alpha}{V^2}\right)(V-\beta)=RT$$

whereas

```
\[
  \left(P+\frac{\alpha}{V^2}\right)(V-\beta)=RT
\]
```

produces

$$\left(P+\frac{\alpha}{V^2}\right)(V-\beta)=RT$$

One may write displayed equations across multiple lines in LaTeX files, so the begin and the end may appear in two different lines.

Handling comments LaTeX does not support multi-line comments. If an (unescaped) % appears anywhere in a line, the rest of the line (including the new-line character at the end) is ignored. Your lex program must ignore whatever appears in a comment. For example, commands or math-mode toggles inside comments must be ignored. The literal % is printed by \%, and this is not to be treated as a comment.

Ignoring other text Your lex program must ignore texts containing none of the above patterns, and will print nothing against these.

Your LEX program

Write appropriate regular expressions and actions for detecting the LaTeX patterns as mentioned above. You may use suitable #define's and global variables at the beginning of your lex program. Compile the lex code with `lex` or `flex`. This will generate the C file `lex.yy.c` as usual.

Your C/C++ program

`#include "lex.yy.c"` in your C/C++ program (henceforth called *your C program*). You may use the compilation flag `-ll` to avoid the error caused by `yywrap()`, or define the function yourself.

Inside your C program, maintain all the commands and environment names that will be encountered during a run. Use a data structure of your own. Any data structure to store and search strings will do. But you must not use any STL data structure. Use one table to store the commands, and another table to store only the environment names (without the \begin or \end, and the braces). Against each string (command or environment name), store a count of how many times the item is used in the LaTeX source. There is no need to keep the tables sorted (with respect to names or the occurrence counts or first/last occurrence instants).

Also, keep two separate counts: one for storing the number of inline mathematical formulas, and the other for storing the number of displayed mathematical formulas.

At the end, print only the overall statistics (see the following sample output). Without these, your program must print nothing.

What to submit

Submit two files: Your lex file (like `latex.l`) and your C file (like `procltx.c` or `procltx.cpp`).

Sample output

The output on the LaTeX file of this document follows. The occurrence counts are shown within parentheses.

```
Commands used:
        \documentclass (1)
        \usepackage (3)
        \renewcommand (1)
        \familydefault (1)
        \sfdefault (1)
        \textheight (1)
        \topmargin (1)
        \textwidth (1)
        \oddsidemargin (2)
        \evensidemargin (1)
        \parskip (1)
        \parindent (1)
        \def (6)
        \dquote (5)
        \tt (15)
        \char (6)
        \$ (13)
        \bs (52)
        \^ (4)
        \_ (5)
        \tilde (3)
        \color (3)
        \bf (6)
        \\ (6)
        \hrulefill (3)
        \vspace (4)
        \bigskipamount (3)
        \LaTeX (16)
        \  (19)
        \bigskip (4)
        \TeX (2)
        \item (9)
        ~ (9)
        \{ (10)
        \} (10)
        \qquad (8)
        \mbox (4)
        \left (3)
        \frac (3)
        \alpha (3)
        \right (3)
        \beta (3)
        \% (3)
        \# (2)
        \it (1)
        \newpage (1)
        \footnotesize (1)
        \input (1)
        \medskipamount (1)
Environments used:
        document (1)
        center (2)
        description (1)
        itemize (1)
        quote (1)
1 math equations found
2 displayed equations found
```