

Wenzhou-Kean University

Project

CPS2231: Computer Organization and Programming

|  |  |
| --- | --- |
| Student Name: | Wang Xinyu |
| Student ID: | 1098648 |
| Project Name: | LaTeX Calculator |
| Lecture: | Dr. Hemn Barzan Abdalla |

Spring Semester-2021

Contents

[1. Introduction 3](#_Toc89271637)

[1.1 Data Structure 3](#_Toc89271638)

[1.2 What is LaTeX 3](#_Toc89271639)

[2. Proposed Problem 4](#_Toc89271640)

[3. Proposed Solution 4](#_Toc89271641)

# Introduction

## Data Structure

The tree data structure in this project is to parse LaTeX mathematical expressions, where the leaf nodes of the tree will store data, and the branch nodes of the tree will be used to store algorithms, or how to process the data.

In besides, I also use Stack to store the digits and symbols that are read while processing LaTeX equations. When a number is read, the number would be stored into the number stack; when a symbol is read, this symbol would be stored into the symbol stack. After finishing building the stack, we finally build the LaTeX mathematical equation tree as I mentioned above by continuously popping and pushing data from the number and symbol stacks.

## What is LaTeX

LaTeX is a document preparation system for TeX typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing. With the typesetting power of LaTeX, it also provides an enormously powerful support for writing mathematical formulas, so that we can easily write mathematical expression on our computers without worrying about the layout ad format. For instance, the following mathematical expression,

could be transform from the LaTeX expression as below

E = \frac{mc^2}{\sqrt{1-\frac{v^2}{c^2}}

And for more complex mathematical expressions, LaTeX also provides excellent tools. For example, for the following mathematical expressions

its LaTeX expression is like:

\begin{aligned}

\nabla \cdot \nabla \psi &= \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} +

\frac{\partial^2 \psi}{\partial z^2} \\

&= \frac{1}{r^2\sin\theta}\Big[\sin\theta \frac{\partial}{\partial r}(r^2\frac{\partial \psi}{\partial r}) +

\frac{\partial}{\partial \theta}(\sin\theta \frac{\partial \psi}{\partial \theta}) +

\frac{1}{\sin\theta}\frac{\partial^2\psi}{\partial \psi^2}\Big]

\end{aligned}

This is the power of LaTeX when dealing with mathematical formulas.

# Proposed Problem

With the power support of mathematical expression in LaTeX, we could easily write elegant and beautiful mathematical equation, however, this raises the problem that when writing mathematical expressions in LaTeX, that is we are not able to copy-and-paste them directly for calculations; we often need to remove the LaTeX identifier, such as \frac, from the expression such that we could calculate it with computer. For example, to calculate the result of which has the LaTeX expression like \frac{1}{2}, we need remove the whole \frac LaTeX identifier from the expression, and add a / sign between 1 and 2, so the LaTeX expression would become 1 / 2 which could easily calculate with computer, or just pasting it into Google, and it would give 0.5 as the result.

# Proposed Solution

Generally speaking, this project or this proposed solution for the problem of calculating mathematical expression in LaTeX could be divided into three layers, the Main layer, the Preprocessor layer, and the Tree Builder layer.

The Main layer is used to interact with the user, it does not handle LaTeX data directly, in short, it acts like the Command Line Interface aka CLI.

The Preprocessor layer performs intermediate data processing, such as cleaning up invalid expressions and completing irregular or simplified equations, to make Tree Builder work properly. In addition, Preprocessor converts some common LaTeX expressions into mathematical expressions, for example, fractional expressions in LaTeX would be converted into division form here.

The Tree Builder layer is the most critical of all layers and is the core component of this LaTeX Calculator. At this layer, the program reads the LaTeX expressions and stores them in the symbolic and numeric stacks, respectively, after which it calls another method for building the LaTeX expression tree to complete and return the tree.

Besides, the LaTeX Calculator is easily extendable, as I have modularized the functional components. For example, if you want the calculator to support factorial operations, which are not yet supported, you can simply inherit from the com.ltc.tree.functions package FunctionNode is an abstract class with a mathematical formula, and then you can add the corresponding mathematical notation to FunctionNodeMapper to make the program support this calculation. The implementation of addition and sine operator in this LaTeX calculator is given below as example to show how simple it would be extended.

public class AdditionNode extends FunctionNode {

public AdditionNode(ValueNode left, ValueNode right) { super(left, right); }

@Override

public double process() { return LeftOperand() + RightOperand(); }

}

public class TriSinNode extends FunctionNode {

public TriSinNode(ValueNode left, ValueNode right) { super(left, right); }

@Override

public double process() { return Math.sin(LeftOperand()); }

}

The Source code of this project could be found in my GitHub Repository 2021-Fall/2232 CPS/2232-FinalProject, check the link below:

https://github.com/WendellXY/2021-Fall/tree/master/2232%20CPS/2232-FinalProject