苏 州 大 学 实 验 报 告

院系	计算机学院	年级专业	21 计科	姓名	方浩楠	学号	2127405048
课程名称	编译原理课程实践					成绩	
指导教师	王中卿	同组实验者	无		实验日期	2023.11.7	

实 验 名 称 基于 PLY 的 LaTex 解析

一. 实验目的

理解并实践将 LaTeX 文档转换为 PDF 的过程。
掌握解析 LaTeX 文件并生成抽象语法树（AST）的技术。
学习如何将 AST 进一步转换为 HTML 和 PDF 格式。
提升 Python 编程和处理文本文件的能力。

二. 实验内容

使用 ply 库来解析 LaTeX 文件并创建 AST。
通过自定义的 Node.py 模块建立和操作 AST。
使用 AST2HTML.py 将 AST 转换为 HTML 格式。
使用 HTML2PDF.py 将 HTML 格式转换为 PDF 文件。
分析并调试转换过程中的潜在问题。

三. 实验步骤和结果

项目结构如下:

```
|--AST/
|   |-- Node.py          # 该模块定义了一个用于表示树结构节点的类 Node
|
|-- AST2PDF/
|   |-- AST2HTML.py      # 该模块用于将 LaTeX 文件转换为 HTML 字符串
|   |-- HTML2PDF.py      # 该模块用于将 HTML 字符串转换为 PDF 文件
|
|--LatexParser/
|   |-- LatexParser.py   # 该模块用于将 LaTeX 文件解析为抽象语法树
|
|-- data/
|   |-- example1.tex     # 示例 LaTeX 文件
|   |-- example2.tex     # 示例 LaTeX 文件
|
|-- output/
|   |-- example1.html    # 输出的 HTML 文件
|   |-- example1.pdf     # 输出的 PDF 文件
|   |-- example2.html    # 输出的 HTML 文件
|   |-- example2.pdf     # 输出的 PDF 文件
|
|-- docs/
|   |--第9次课.docx       # 实验要求
|
|-- run.py               # 顶级脚本，用于运行项目
|-- README.md            # readme文件，项目概述和使用说明
|-- requirements.txt     # 项目依赖项说明
```

项目运行方式:
要运行此项目，需要安装以下依赖项：

```
ply~=3.11
pytest~=7.4.3
fpdf2~=2.7.6
```

您可以通过运行以下命令来安装这些依赖项：

```
pip install -r requirements.txt
```

使用方法

方式 1
使用命令行参数指定要转换的文件路径，例如：

```
python run.py data/example1.tex output/example1.pdf output/example1.html
```

这条指令会读取`data/example1.tex`文件，并将其转换为`output/example1.pdf`和`output/example1.html`文件。

只要是形如

```
python run.py {读取的 latex 文件位置} {输出的 pdf 文件位置} {输出的 html 文件位置}
```

的指令均可以运行
其中 html 文件作为中间文件，可以不指定路径.这样就不会生成 html 文件,而是直接将 latex 文件转换为 pdf 文件

方式 2
也可以直接打开 run.py,然后修改 run.py 的这一部分:

```
if __name__ == "__main__":
    tex_filename = ""
    pdf_output_filename = ""
    html_output_filename = ""
```

修改这一部分也可以实现转换功能

标签实现情况如下:

\begin{document}...\end{document}	实现
\title	实现
\author	实现
abstract	实现
\section	实现
\subsection	实现
itemize	实现
item	实现

AST 的生成情况:
对于 example2.tex:

```
+ [DOC]
  + [CONTENT]
    + [TITLE]
      + How to Structure a Latex Document
    + [AUTHOR]
      + Andrew Roberts
    + [ABSTRACT]
      + In this article, I shall discuss some of the fundamental topics in producing a structured document.  This document itself does not go into much depth, but is instead the output of an example of how to implement structure. Its Latex source, when in used with my tutorial provides all the relevant information.
    + [SECTIONS]
      + [SECTION](Introduction)
        + This small document is designed to illustrate how easy it is to create a well structured document within Latex.  You should quickly be able to see how the article looks very professional, despite the content being far from academic. Titles, section headings, justified text, text formatting etc., is all there, and you would be surprised when you see just how little markup was required to get this output.
      + [SECTION](Structure)
        + One of the great advantages of latex is that all it needs to know is the structure of a document, and then it will take care of the layout and presentation itself.  So, here we shall begin looking at how exactly you tell latex what it needs to know about your document.
      + [SUBSECTION](Top Matter)
        + The first thing you normally have is a title of the document, as well as information about the author and date of publication. In latex terms, this is all generally referred to as top matter.
      + [SUBSECTION](Author Information)
        + It is common to not only include the author name, but to insert new lines after and add things such as address and email details.  For a slightly more logical approach, use the AMS article class and you have the following extra commands:
        + [ITEMIZE]
          + [ITEM]
            + The author's address.  Use the new line command
          + [ITEM]
            + Where you put any acknowledgments.
          + [ITEM]
            + The author's email address.
          + [ITEM]
            + The URL for the author's web page.
        + The first thing you normally have is a title of the document, as well as
```

information about the author and date of publication. In latex terms, this is all generally referred to as top matter.

由于在 HTML 中,section 和 subsection 层次相同,因此在 AST 中,section 和 subsection 也可以归在同样的层次中.
原先给出的生成 AST 的算法中,sections 使用了递归,导致树的结构不够清晰.因此我进行了修改,使得所有的 section 和 subsection 都在同一层次

将 Latex 转为 HTML 的函数:

```
def Node2Html(node: Node) -> str:
    """
    将 Node 对象转换为 HTML 格式的字符串。

    Args:
        node: Node 对象。用来表示 Latex 文档的 AST

    Returns:
        HTML 格式的字符串。
    """
```

转换 title;

```
f"""<div class="title">{section_node.getChildren()[0].getData()}</div>\n"""
```

转换 AUTHOR:

```
f"""<div class="author">{section_node.getChildren()[0].getData()}</div>\n"""
```

转换 section:

```
<div class="section">
    <h2>{current_section_title}</h2>
        {current_section_content}
</div>
```

转换 subsection:

```
<div class="subsection">
    <h3>{current_section_title}</h3>
        {current_section_content}
</div>
```

转换 ITEM 和 ITEMIZE:

```
{items_html}
<li>{list_item.getChildren()[0].getData()}</li>
```

```
{current_section_content}
<ul>
    {items_html}
</ul>
```

example2.tex 转换后的 HTML:

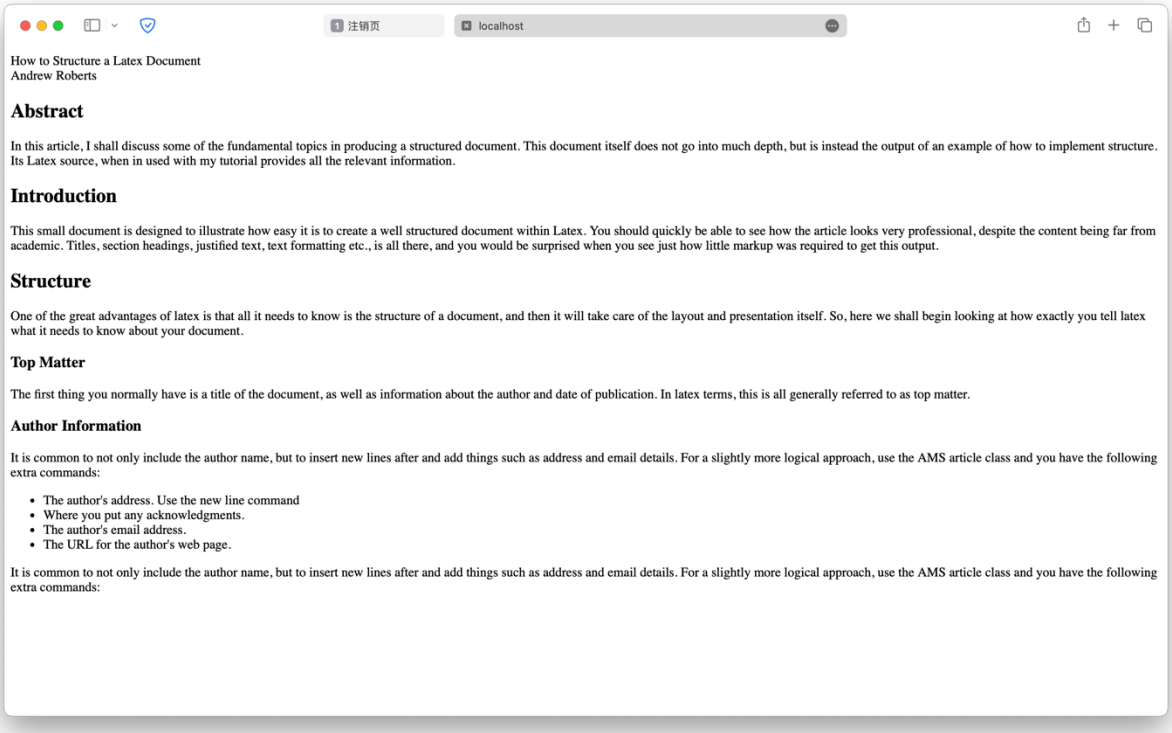
```
<!DOCTYPE html>
<html>
<body>
<div class="title">How to Structure a Latex Document</div>
<div class="author">Andrew Roberts</div>

    <div class="abstract">
        <h2>Abstract</h2>
        <p>
            In this article, I shall discuss some of the fundamental topics in
producing a structured document.  This document itself does not go into much depth,
but is instead the output of an example of how to implement structure. Its Latex
source, when in used with my tutorial provides all the relevant information.
        </p>
    </div>

    <div class="section">
        <h2>Introduction</h2>


        <p>
            This small document is designed to illustrate how easy it is to create a
well structured document within Latex.  You should quickly be able to see how the
article looks very professional, despite the content being far from academic.
Titles, section headings, justified text, text formatting etc., is all there, and you
would be surprised when you see just how little markup was required to get this
output.
        </p>

    </div>

    <div class="section">
        <h2>Structure</h2>


        <p>
            One of the great advantages of latex is that all it needs to know is the
structure of a document, and then it will take care of the layout and presentation
```

```
itself.  So, here we shall begin looking at how exactly you tell latex what it needs
to know about your document.
        </p>

    </div>

    <div class="subsection">
        <h3>Top Matter</h3>



        <p>
            The first thing you normally have is a title of the document, as well as
information about the author and date of publication. In latex terms, this is all
generally referred to as top matter.
        </p>

    </div>

    <div class="subsection">
        <h3>Author Information</h3>




        <p>
            It is common to not only include the author name, but to insert new lines
after and add things such as address and email details.  For a slightly more logical
approach, use the AMS article class and you have the following extra commands:
        </p>

        <ul>



                            <li>The author's address.  Use the new line command </li>
                            <li>Where you put any acknowledgments. </li>
                            <li>The author's email address. </li>
                            <li>The URL for the author's web page. </li>
        </ul>

        <p>
            It is common to not only include the author name, but to insert new lines
```

```
after and add things such as address and email details.  For a slightly more logical
approach, use the AMS article class and you have the following extra commands:
        </p>

    </div>
    </body>
</html>
```

可以发现 example2.tex 中的标签均成功匹配

在浏览器中查看 example2.html



发现 Latex 文件中的各种标签均被匹配,生成的 HTML 文档格式正确

最后使用 FPDF 来实现将 HTML 转换为 PDF 文件
用来实现转换的函数:

```
def Converter(tex_filename: str, pdf_output_filename: str, html_output_filename: str
= "") -> None:
    """
    将 tex 文件转换为 pdf 文件
    Args:
        tex_filename: tex 文件的路径
        pdf_output_filename: 输出的 pdf 文件的路径
        html_output_filename: 输出的 html 文件的路径
```

```
    Returns:
        None
    """
```

最终生成的 example2.pdf

How to Structure a Latex Document Andrew Roberts

Abstract

In this article, I shall discuss some of the fundamental topics in producing a structured document. This document itself does not go into much depth, but is instead the output of an example of how to implement structure. Its Latex source, when in used with my tutorial provides all the relevant information.

Introduction

This small document is designed to illustrate how easy it is to create a well structured document within Latex. You should quickly be able to see how the article looks very professional, despite the content being far from academic. Titles, section headings, justified text, text formatting etc., is all there, and you would be surprised when you see just how little markup was required to get this output.

Structure

One of the great advantages of latex is that all it needs to know is the structure of a document, and then it will take care of the layout and presentation itself. So, here we shall begin looking at how exactly you tell latex what it needs to know about your document.

Top Matter

The first thing you normally have is a title of the document, as well as information about the author and date of publication. In latex terms, this is all generally referred to as top matter.

Author Information

It is common to not only include the author name, but to insert new lines after and add things such as address and email details. For a slightly more logical approach, use the AMS article class and you have the following extra commands:

- The author's address. Use the new line command
- Where you put any acknowledgments.
- The author's email address.
- The URL for the author's web page.

It is common to not only include the author name, but to insert new lines after and add things such as address and email details. For a slightly more logical approach, use the AMS article class and you have the following extra commands:

发现 Latex 文档已经被成功转换为了 PDF

四. 实验总结

通过本次实验，我加深了对 LaTeX 文档结构的理解，学会了如何使用 Python 来解析和转换文件格式。遇到的挑战包括解决依赖项冲突、调试 AST 转换中的错误，以及优化 HTML 到 PDF 的布局转换。实验过程中，我学会了更高效地调试代码，同时也认识到了代码的模块化和文档编写的重要性。对于未来的工作，我计划改进错误处理机制，并提高转换工具的用户友好性和健壮性。