Assignment 2

Transformed Grammar

The transformed grammar is in `./final.grm`. This grammar is an LL(1) grammar, and this can be verified using `./final.grm.ucalgary` on the University of Calgary website.

First Follow Sets

The first-follow sets of the transformed grammar are stored in `./first_follow.txt` in a readable format.

Design

I chose to take the *table-driven* approach, where I applied a simple parsing algorithm on a *parsing table*(generated using the Calgary website). This approach allowed me to fix bugs in my grammar quickly because changes in the grammar did not affect the parsing algorithm. My implementation of table driven parser is can be seen in the recursive function called `parse_helper()` in `parse.py`

I added a `eof`(end of file) terminal to my grammar which allows me to simplify my table parsing algorithm.

I manually removed:

- **First Set ambiguities**: these were relatively simple to remove
- **First and Follow Set Clash abiguities**: I found these very hard(if not impossible) to remove on `statement` and `factor` non-terminals. Hence, I ended up simplifying the grammar to remove these. More precisely, my transformed grammar currently does not allow long chains of function calls like `x().y().z()` due to simplification.

Use of Tools

Rust and Python Unfortunately, for now, I have implemented my lexer in the **Rust** programming language and my table-driven parser in **Python**. All the lexer code is located in `./src` and the parser code is in `./parser.py`.

University Of Calgary Website This website came is very handy for generating the parsing table and removing abiguities from my grammar.

Submission

- **Source Code** : Lexer source code is in `./src` and parser source code is in `parser.py`.
- **Test source files**: All test source files are in `./test_cases/test_source_files`
- **Test derivations** : All derivations of each test file is `./test_cases/test_syntax_derivations`
- **Driver** : The driver is `./parserdriver.py`

Instructions on using the Driver

- Install RustLang then run following in terminal in this folder:

```
cargo build
```

- Install the following three python libraries using the following three commands in terminal:

```
pip install bs4
pip install lxml
pip install ipython
```

- Place any new test source file in `./test_cases/test_source_files`(see example files in folder) and execute driver using `python parserdriver.py` in terminal.
- The driver will parse ALL `.src` files in `./test_cases/test_source_files`.
- You will find corresponding generated tokens in `./test_cases/test_tokens_files`.
- You will find corresponding generated derivations in `./test_syntax_derivations`.