CS-335

README

MILESTONE 1

Group 12

Aditya Ajmera (210056)
Maurya Aryan Swaminath (210595)
Depanshu Sahu (210316)

1 Tools Used

- We have used the lexical specifications and the grammar for Python 3.8.
- We have used the GraphViz tool to visualise AST for our grammar.
- We have written the lexer part in Flex and the grammar part in Bison.

2 Compilation Instructions

After extracting the zip, open the terminal in the folder and execute:

```
cd milestone1/src
```

Now we are in the same directory as the make, lexer and parser files. Now run

```
make clean
```

The above command cleans up the previously created files by lexer and parser.

```
make compile
```

This command compiles the lexer and parser and makes the `main.exe` file.

3 Command Line Options

- **- -input** Add this flag to specify an input file to the parser. This is a required flag.
  Example:

  ```
  ./main --input=input.py
  ```

- **- -output** Add this flag to specify the output dot file to the parser which will contain the dot code to generate AST. This is optional and default output is `output.dot`.
  Example:

  ```
  ./main --input=input.py --output=result.dot
  ```

- **- -help** Use this flag to know about the rules to run the commands.
  Example:

  ```
  ./main --help
  ```

- **- -verbose** Use this flag to apply the debug mode of bison parser. This is an optional flag.
  Example:

  ```
  ./main --input=input.py --output=result.dot --verbose
  ```

4 Execution Instructions

To execute, run the following commands:

```
./main --input=<input_file_name> --output=<output_file_name>
```

For example:

```
./main --input=input.py --output=result.dot
```

Now generate the AST pdf from dot file using:

```
dot -Tpdf <output_file_name> -o AST.pdf
```

In our example case `<output_file_name>` is replaced by `result.dot`:

```
dot -Tpdf result.dot -o AST.pdf
```