Python Compiler - Milestone 2

Akshat Gupta [*] Devansh Kumar Jha [†] Shashwat Gupta [‡]

April 3, 2024

1 Compilation Instructions

To build this repository follow the following steps:

```
cd src          # Go the the source code directory
make all        # Run the Makefile to compile the toolchain
./cs335 --help  # Get information about the functioning of the compiler
./cs335 --input <path to input file>
make clean      # Clean all the compiled binaries
```

For more specific output you can also run the following commands:

```
# Give input directly from the terminal. Use --verbose or --full to display debug output.
./cs335

# Redirect debug outputs to the output file.
./cs335 --input <path to input file> --output <path to output file> --verbose
./cs335 --input <path to input file> --output <path to output file> --full

# Generate Symbol Table dump for the input program
./cs335 --input <path to input file> --symbol <path to directory of symbol table dump>

# Generate 3AC text file for the input program
./cs335 --input <path to input file> --3ac <path to store 3AC text file>

# Generate AST pdf and DOT script for the input program
./cs335 --input <path to input file> --dot <path to store dot script>
./cs335 --input <path to input file> --ast <path to store pdf file for AST>
./cs335 --input <path to input file> --dot <path to store dot script> --ast
<path to store pdf file for AST>

# Generate Parse Tree pdf for the input program
./cs335 --input <path to input file> --parse <path to store pdf file for parse tree>
```

[*]Fourth Year Undergraduate, IIT Kanpur, akshatg20@iitk.ac.in
[†]Fourth Year Undergraduate, IIT Kanpur, dkjha20@iitk.ac.in
[‡]Fourth Year Undergraduate, IIT Kanpur, shashwatg20@iitk.ac.in

NOTE:

- We have provided some testcases in the *tests* subdirectory. To test the *./cs335* binary on these testcases, use the command:

```
./cs335 --input ../tests/test1.py
```

- The following condition should be met by every testcase: **Input test cases will use two spaces for indentation and will have a final newline.**
- While using –**ast** and –**dot** flags, the path to store this files should at the very least include filename. For example, if you want to generate the file **ast.pdf** in the same directory as the binary, use the command:

```
./cs335 --input <path to input file> --ast ./ast.pdf
```

- The use of –**ast** flag generates the AST pdf and **halts** further phases of compilation, like semantic analysis or 3AC generation. To generate symbol table dump, directly use the –**symbol** flag. To generate the 3AC text file, directly use the –**3ac** flag.
- When you set the –**3ac** flag, the flags –**ast** and –**parse** are ignored.
- Symbol table for **each function and class** would be dumped in a **separate csv file**.
- We recommend that you store all the outputs in specific folders in the **milestone2/output** subdirectory.
- For example, use the following command for generating symbol table for *test1.py*:

```
./cs335 --input <../testcases/test1.py> --symbol <../output/symboltable>
```

- For example, use the following command for generating 3AC for *test1.py*:

```
./cs335 --input <../testcases/test1.py> --3ac <../output/3ac/test1.txt>
```

2 Different flag functiol table dump:nalities

2.1 help

```
./cs335 --help
```

Display of information about the compiler flags.

2.2 input

```
./cs335 --input <path to input file>
```

Adds the path of the input file. By default it is the standard input. The next argument after –input flag should correspond to the location where the input file is present.

2.3 output

```
./cs335 --input <path to input file> --output <path to output file>
```

Adds the path of the output file. By default compiler output is displayed at the standard output. The next argument after –output flag should correspond to the location where the output is to be stored.

2.4 verbose

```
./cs335 --input <path to input file> --verbose
```

Prints detailed compilation log and debug outputs.

2.5 full

```
./cs335 --input <path to input file> --full
```

Prints the complete debug output.

2.6 error

```
./cs335 --input <path to input file> --error <path to error file>
```

Adds the path of the error file. By default it is the standard error. The next argument after –error flag should correspond to the location where the error is redirected.

2.7 debug

```
./cs335 --input <path to input file> --debug <path to debug file>
```

Adds the path of the debug output file. By default it is the standard output. The next argument after –debug flag should correspond to the location where the debug output is redirected.

2.8 ast

```
./cs335 --input <path to input file> --ast <path to store pdf file for AST>
```

Configures the compiler to output Abstract Syntax Tree of the input program in a PDF. The next argument after –ast flag should correspond to the location where the ast file is to be stored.

2.9 dot

```
./cs335 --input <path to input file> --dot <path to store dot script>
```

Configures the compiler to output the DOT script corresponding to input program in a .dot file. The next argument after –dot flag should correspond to the location where the dot file is to be stored.

2.10 parse

```
./cs335 --input <path to input file> --parse <path to store pdf file for Parse Tree>
```

Configures the compiler to output the Parse tree of the input program in a PDF. The next argument after –parse flag should correspond to the location where the parse tree file is to be stored.

2.11 symbol

```
./cs335 --input <path to input file> --symbol <path to directory of symbol table dump>
```

Configures the compiler to dump details of information stored in symbol table in form of .csv file outputs. The next argument after –symbol flag should correspond to the directory where the symbol table is to be dumped.

2.12 3ac

```
./cs335 --input <path to input file> --3ac <path to store text file for 3AC>
```

Configures the compiler to print the output 3AC IR of the source program in a text file. When this flag is set the flags –ast and –parse will be ignored. The next argument after –3ac flag should correspond to the location where the 3AC IR text file is to be stored.