

# CS335 - Milestone 2

Pranjal Singh, Dev Gupta, Deven Anil Gangwani

March 2024

## 1 Introduction

This document describes our implementation of milestone 2 of the course project of CS335 (The Manhatthon Project). Our parser recognises a statically typed subset of Python 3.12 and generates 3 AC instructions corresponding to the input program.

By default, 3AC instructions are printed to `tac.txt` and the symbol table is printed to `symbol_table.csv`. Lexical analysis and parsing of the input were implemented in milestone 1.

## 2 Compilation Commands

### 2.1 List of Commands

- Generating the parser (`parser.tab.c`) and header file for the lexer (`parser.tab.h`):  
`$ bison -d parser.y`
- Generating the lexer (`lex.yy.c`):  
`$ flex lexer.l`
- Compilation: (the `-lfl` flag may be needed)  
`$ g++ -o parser parser.tab.c lex.yy.c`
- Running the parser with input redirection (generates `tac.txt` and `symbol_table.csv`):  
`$ ./parser -input input.py`

### 2.2 Makefile Usage

To simplify the compilation procedure during programming and debugging, we used GNU `make`. We have defined the following targets:

- `$ make`: Same as `make parser`.
- `$ make parser` Compiles the lexical and grammar specifications into an executable (`parser`)
- `$ make test` Compiles `input.py`, generates `temp.pdf` and `ast.dot` and then cleans up.
- `$ make clean` Deletes intermediate files.

Additionally, we used the GNU C debugger during testing, for which we included `-g` flag in the `g++` command.

### 3 Command-line Arguments

- `-input input.py`: Use `input.py` as the input. By default, the lexer reads from the console (`stdin`). Input redirection can also be used.
- `-output output.txt`: Create/overwrite `output.txt` with the 3 Address Code of the program.
- `-help`: List command-line options and exit.
- `-verbose shift`: Filter shift operations from the inbuilt `bison` traces and prints to `stderr`. Useful to figure where an error is encountered.
- `-verbose reduce`: Filter and print reduce operations.
- `-verbose sr`: Filter and print shift and reduce operations.
- `-verbose all`: Copy the entire trace. (Runs into thousands of lines.)
- `-verbose srls`: Filter and print shift and reduce operations and lookahead token in each step. Exclude stack state.

Error messages in semantic analysis are printed to the console with line numbers directly.