CS335 : Milestone1 Documentation

Roll no.- 210750, 210557, 210543

March 3, 2024

1 Introduction

This document provides an overview of the tools used, compilation instructions, and execution instructions for the project.

2 Tools Used

The following tools were used in the project:

- **Bison (parser.y)**: Bison is a parser generator used to generate the parser from the grammar file `parser.y`. It converts the grammar rules into a C++ parser (`parser.cpp`) along with the header file (`parser.hpp`).
- **Flex (lexer.l)**: Flex is a lexical analyzer generator used to generate the lexer from the lexical specification file `lexer.l`. It converts the regular expressions defined in `lexer.l` into a C++ lexer (`scanner.cpp`).
- **GNU Compiler Collection (g++)**: The GNU Compiler Collection is used to compile the source files into an executable. It compiles the generated parser (`parser.cpp`), lexer (`scanner.cpp`), and any additional source files (e.g., `ast.cpp`) into the executable named `parse`.
- **Graphviz** : Graphviz is a graph visualization software used to visualize the parse tree. The parse tree is generated as a DOT file named `output_file.dot`. It can be converted into an PDF format using the `dot` command.

3 Compilation Instructions

To compile the project, use `make` command.It involves the following commands:

1. Generate the parser using Bison:
   `bison -d -o parser.cpp parser.y`
2. Generate the scanner using Flex:
   `flex -o scanner.cpp lexer.l`
3. Compile the source files:
   `g++ parser.cpp scanner.cpp ast.cpp -o parse -std=c++11`

4 Execution Instructions

To execute the compiled program, following command line tools are supported:(Note: Two files need be provided always input and output.There is no default file.)By default we can execute using `./parse path-to-input-file path-to-output-file`

1. -input:
   `./parse -input path-to-input-file path-to-output-file`
2. -output:
   `./parse -output path-to-output-file path-to-input-file`
3. -help:
   `Gives instructions to execute the file`
4. -verbose:
   `Prints statewise execution of parser,implemented using yydebug in Bison`
5. visualize the parse tree using Graphviz:
   `dot -Tpdf output_file.dot -o graph.pdf`