Changes After Milestone 2

We have removed sqrt and log function from our grammar.

Analysis of conflicts

Our compiler implementation has only one shift-reduce conflict and zero reduce-reduce conflicts. The shift-reduce conflict present in our implementation is
WARNING: shift/reduce conflict for ELSE in state 341 resolved as shift
After rigorous analysis, we have concluded that this conflict can't be resolved. According to our understanding, this shift-reduce conflict occurs when the if statement ends, then it can either reduce if or it pushes else into the stack. Since we must check for an else each time, there should be a shift operation. Thus, this conflict can't be removed, and the default behavior, i.e., a shift should be done in this case.

Output of *git show parser*

```
—--------------------------------------------------
Author: Ketan Chaturvedi <ketanchaturvedi.24@gmail.com>
Date:   Fri Feb 18 23:09:09 2022 +0530

      Updated README wrt parser and added dot file

diff --git a/README.md b/README.md
index 62c20e9..d4bc4ce 100644
--- a/README.md
+++ b/README.md
@@ -10,6 +10,25 @@ This repository contains a C compiler made as part of
compiler course (CS335). T
 3. Priyanshu Yadav (190652)
 4. Hardik Sharma (190353)

+## How to Run
+To run the parser, cd into the compiler directory and use the following
command:
+```
+python3 src/parser.py <test file path>
+```
```

```
+The above command will tell whether the input file is accepted by grammar or
not and generates a .dot file in *src* directory. Now to generate Parser
Automata, use the following command:
+```
+sfdp -x -Goverlap=scale -Tpng tmp/try2.dot > tmp/graph_out.png
+```
+-------------------------------------------------------------------------
---
+To run the lexer, cd into the compiler directory and use the following
command:
+```
+python3 src/lexer.py <test file path>
+```
+
 ## Milestones
 1. **Milestone 1** : Added compiler source language specifications
-2. **Milestone 2** : Built lexer in python to tokenize C code
\ No newline at end of file
+2. **Milestone 2** : Built lexer in python to tokenize C code
+3. **Milestone 3** : Developed parser for the source language that outputs
the Parser Automaton in a graphical form
+     - Wrote grammer rules for the C language
+     - Removed several reduce/reduce and shift/reduce conflicts
+     - Used action table and goto table to generate .dot file which when
processed by Graphviz, gave a Parser Automaton image file
\ No newline at end of file
diff --git a/src/graph_file.dot b/src/graph_file.dot
new file mode 100644
index 0000000..049163d
--- /dev/null
+++ b/src/graph_file.dot
@@ -0,0 +1,3876 @@
+digraph DFA {
+     0 [label="I0"];
+     1 [label="I1"];
+     2 [label="I2"];
+     3 [label="I3"];
+     4 [label="I4"];
+     5 [label="I5"];
+     6 [label="I6"];
+     7 [label="I7"];
```