Annotating Java Programs to Provide Feedback to CS1 Students

Krish Godiawala
Advisor: Dr. Carlos Rivero

Introduction

Often mistakes made by CS1 students are repetitive, for example forgetting semicolons, incorrect loop iterations to name a few. Graders and professors alike are left providing the same feedback to them which is a monotonous task.

CS1 students often get feedback from professors and graders about their code which at times they fail to understand, or worse fail to implement those suggested changes. Providing a student with feedback such as incorrect loop iteration is far less intuitive than providing them with actual code which he/she can use.

Often students misplace their assignments and do not have their old mistakes to look up when they really need to.

Background

An Abstract Syntax Tree (AST) is a tree representation of source code with each AST node corresponding to a name, type, expression, statement or declaration[1]. The figure below is sample source code, and its corresponding AST. Each node in the AST either represents a variable, operator or constant.

Codepuorii[1] is an online tool that was created to teach programming languages to students. It allowed one to have dynamically changing text depending upon the state of the program by annotating its AST's.

![Sample AST](image)

\[
x := a + b; \\
y := a * b; \\
while (y > a) \\
\{ \\
\quad a := a + 1; \\
\quad x := a + b \\
\}
\]

![Program Tree](image)

System Design

We have developed a system that allows professors and graders to provide feedback to CS1 student code written in Java. The system also provides a way to suggest code which is used to generate a new program file with the changes.

![System Overview](image)

**System Overview**

**User Interface**: It is a text editor where the program is loaded and graded.

**Backend System**: The User Interface communicates with the backend system to notify of any changes and requests. These requests include loading the student program, adding new feedback, manipulating source code.

**AST Handler**: Responsible for performing AST modifications and selecting the correct AST subtree using Lowest Common Ancestor.

**Data Store**: The data store is where submissions, graded assignments and feedback is stored.

![Feedback Box](image)

Provide feedback by selecting the construct to annotate which opens a dialog box with a list of old feedback.

Results

The annotated feedback is visible on the user interface, where it is underlined and an image is visible to the left of it which on mouse hover shows the feedback.

![Annotated Feedback](image)

- The system successfully allows graders and professors to provide feedback to CS1 students.
- The students can see all the feedback provided on the user interface in an easy to understand manner.
- They now have a correct program to directly compare their code with and see what they did wrong.
- Students have their graded assignments available electronically for the future.

Future Work

- Put the system into circulation amongst students and professors to take their feedback and integrate it into the system.
- Improve the system by auto detecting errors which can be done using subgraph matching on AST's.
- Using a centralized data store like a NoSQL database to store feedback rather than storing serialized results into the filesystem.
- Add support to provide feedback for a project having multiple packages and classes rather than only supporting standalone classes.

Contact

Krish Godiawala
https://github.com/krishtgodiawala/Capstone
ksg8418@rit.edu

References

2. Codepuorii: https://github.com/krishtgodiawala/Capstone
3. Sample AST: https://csuytech.files.wordpress.com/2008/10/circular.png