CSC 311 Fall 2017

Milestone #3

Due Nov. 19, 2017 11:59PM

Instructions

Submit your completed work using Blackboard before the deadline.

Submit the **JFlex and Byacc/J file** and **also the inputs you used for testing**.

Project Description and Requirements

The goal of this milestone is to perform additional semantic analysis where it can use **type** information to check for type mismatches.
Start the milestone by first downloading the files provided.
The following are the new functionality you are to implement.

1. Add a **DEBUG** mode. Only when debug mode is ON, print tree and symbol table.
    a. Ex) java Parser -d input.txt (this will print the tree and symbol table)
    b. Ex) java Parser input.txt (this will only show compiler errors)

2. Modify Jflex file to add type information when **creating a node for INT, FLOAT, and STRING**. INT is defined as any number of integers (123). FLOAT is defined as any number of integers followed by a decimal point followed by any number of integers (123.123). STRING is defined as any alpha-numeric characters inside double quotes ("abc123").
    a. Look at Node.java and notice that it now has a "type" field to store data type of this node.

3. **Assign a type for each node**. The type is one of INT, FLOAT, and STRING. For operation nodes ("+" or "-"), you do not need to assign a type. We will assume the operations are permitted on all types. So simply ignore the operation nodes when type checking.
    a. To actually perform a type check, one method is to do a post-order walk to find all nodes and its type. The children nodes should all have matching type.
    b. You should maintain a symbol table with the variable name (key) and its type (value). I would recommend keeping a separate symbol table instead of re-using existing one, for better debugging purposes.

4. Be able to do simple **error recovery**, meaning don't throw any exception. You must continue and find all errors.

Grading Criteria

1. Can assign type information to all nodes – 20%
2. Can store and retrieve type information using symbol table – 20%
3. Implemented DEBUG mode – 15%
4. Can catch type mismatches and print the error – 30%
5. Can do error recovery, and show all compile errors – 15%