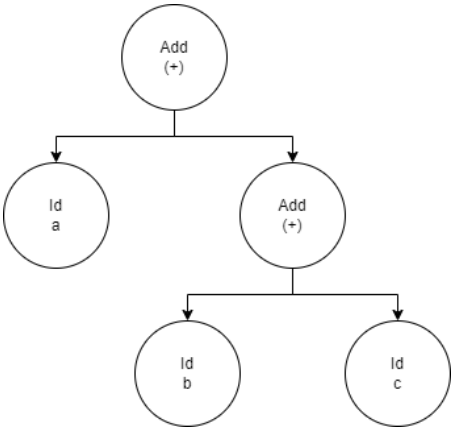
CSC 345 Lab – Code Generation - Coding

Overview

Write Java code that will traverse a simple AST and generate assembly code.

Here is the AST tree:



AST Class

Create a class named AST.

- Member variable. Create a member to store a list of strings. This will be used to hold the lines of assembly code. Initialize it to a new instance of the collection.
- Member variable. Create a member variable that contains a number for the next open register. Initialize it to 1.
- Member variable. Create a member variable that holds a reference to the AST nodes. This is basically the root node of the AST. Make the type ASTExprBase (you will need to define this ASTExprBase class first, see below). This variable should be initialized in the default constructor (see below for specifications).
- Method. Add a method to get an open register. This method should return a string such as r1, r2, etc. Make sure to increment the open register member variable.
- Define the following classes inside of AST.
    - ASTExprBase. This is an abstract base class for expressions. Add the following abstract method:
      abstract String code();
    - ASTId. Should inherit from ASTExprBase.
        - Member variable. Create a member variable to store the id name.
        - Constructor. Add a parameterized constructor to initialize the member variable.
        - Method. Override the code method. It should generate assembly code to load the variable into a register. It should return the register it loads the variable in.

    - ASTAdd. Should inherit from ASTExprBase.
        - Member variable. Create a member variable for the left side of the add. The type should be ASTExprBase.
        - Member variable. Create a member variable for the right side of the add. The type should be ASTExprBase.
        - Constructor. Add a parameterized constructor to initialize the member variable.
        - Method. Override the code method. It should generate new lines of assembly code to perform an add. The assembly code should be added to the AST member variable collection that stores code. Return the register that has the result of the add.
- Method. Add a method to generate code from the AST. It should first call code() on the root of the AST. It should concatenate all lines of code from the code collection and return it as one string. There should be a newline character at the end of each line of code. Here is the method header:
  String generateAssemblyCode()
- Method. Default Constructor. In the AST default constructor, create instances of AST nodes as necessary to create the tree shown in the picture at the start of this document.

Main Class

In the main method create an instance of AST and display the code it generates. Here is some sample output:

```
Assembly Code
load r1, a
load r2, b
load r3, c
add r2, r3, r4
add r1, r4, r5
```