Minijava
Type Checking & Intermediate Code Generation

Introduction

This project completed in March – May of 2020 , in the context of "Compilers" course. The purpose of the project was in first place the type checking of Minijava . The second part of that project was the Intermediate Code Generation .

Methodology

The first part is to use the two .jar files in the src/jar directory to produce the visitors form the Minijava grammar . This is made from the makefile.

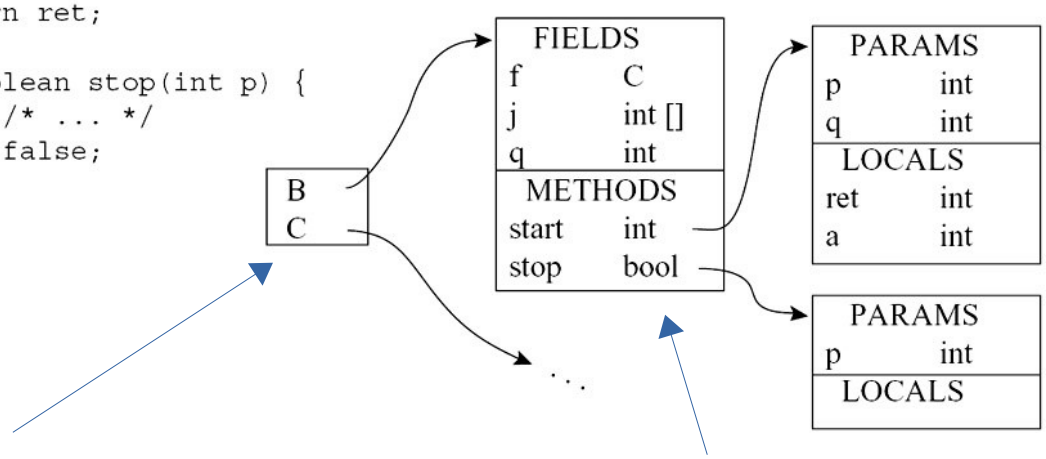
Type Checking :

In Type Checking I created two visitors . The first visitor (file visitors/DeclVisitor.java) scans the minijava program to find declarations of classes , methods and variables and push these informations in the Symbol Table.

For the Symbol Table I followed the below logic :

```
class B {
    C f;  int [] j;   int q;
    public int start(int p, int q) {
         int ret;   int a;
         /* ... */
         return ret;
    }
    public boolean stop(int p) {
              /* ... */
        retun false;
    }
}

class C {
   /* ... */
}
```



The Symbol Table has a record for every class . Each Class Record has

a data structure which keeps the fields of the class and another data structure which keeps the variables of the class . Each Method Record has a data structure which keeps the parameters of the method and another data structure that keeps the local vairables of the method.

Class Record → File types/ClassInfo.java
Method Record → File types/MethodInfo.java
Field Record → File types/FieldInfo.java

The Second Visitor (file visitors/TypeCheckVisitor.java) scans the Minijava Program and checks the types of each expression , using the information of the Symbol Table from the previous Visitor . Each Sub-Expression returns its type to the father Expression and continues from the lowest to the highest by checking the types to be valid in each step.

Code Generation :

I generate code for LLVM . So the output program exmaple.ll (for example if the input program was called example.java) should be compiled with clang to be executed,
In the end of this README it is mentioned how to compile with clang.

The Code Generation it is produced by a visitor (file visitors/GenCodeVisitor.java) . I redirected the stdout to the *.ll file .

Conclusion

To use my Minijava Type Checking and Code Generation you should give these commands in command line:
(Supposed the make is already done)

> java Main Example.java (example java file)
> clang-4.0 -o out Example.ll
>./out

Georgios Liakopoulos