CS 565, Jens Palsberg Oct 22, 1998

Homework 6: Java Flow-Directed Inlining

A grammar for MiniJava in JavaCC notation is on the CS 565 webpage. The goal of the homework is to write a MiniJava optimizer which inlines as many message sends as possible/reasonable. The idea is to do flow analysis in the sense of OOTS Chapter 5 (in which flow analysis is known as "type inference"), and then use the computed information as the basis for replacing message sends with possibly conditional statements where each branch is inlined code. There are many possible strategies for doing this, and your task is to invent one and implement it.

Ideally, the output language would be machine code. To save you from the trouble of writing all of an optimizing compiler, the output language is Java where you can take advantage of the "instanceof" expression to test the class of an object.

So, use JTB and JavaCC and write in Java one or more visitors which transforms a MiniJava program to a Java program in such a way that some message sends get inlined.

You can assume that the MiniJava program type checks. If you decide not to handle all of MiniJava, then state explicitly what parts you cover.

Your main file should be called `Optimize.java`, and if `Prg.java` contains a program to be optimized, then

```
java Optimize < Prg.java > OptimPrg.java
```

creates the program `OptimPrg.java` which has the same behavior as `Prg.Java`, and which is like `Prg.Java`, except that some message sends have been inlined.

Submit your program together with a README file which describes in detail your strategy for inlining. Grading will be based mostly on what works, and partly on how smart and elegant your inlining strategy is.