COMP207P

Assessment 2

Report

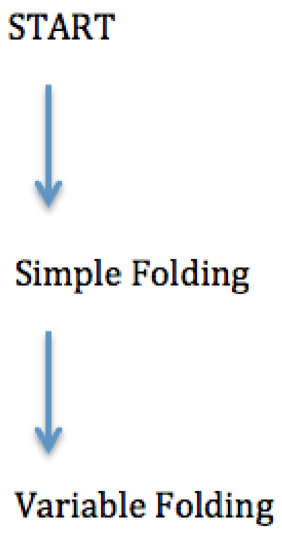
Da An
Enze Liu
Jiahui Gu

Introduction

The program can do simple folding (task 1) and constant variable folding (task 2) and dynamic folding (task 3). However, the program can not optimise the program which contains loops. It can pass all the test in the sample test, except for method four in DynamicVariableFolding.java.

Algorithm

The basic structure of the program is do constant folding in constant pool (task 1) in the method *simpleFolding()*, then perform the rest optimisations in variableFolding(). Which looks like this:

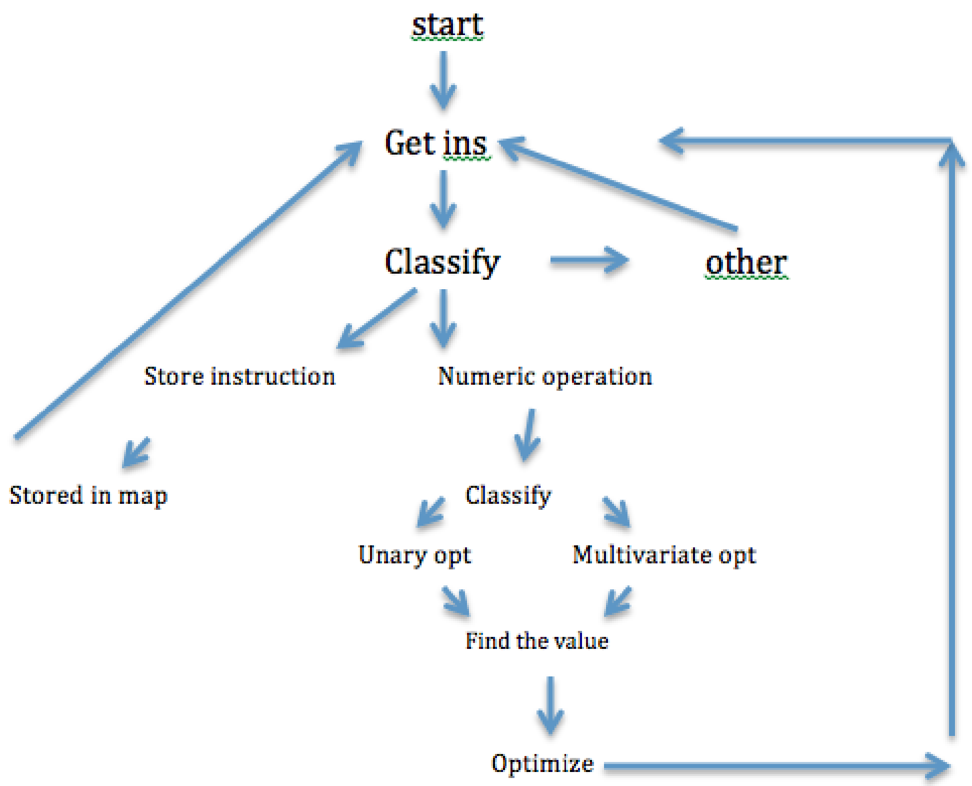


1. Simple Folding

   The method will go through each instruction and look for pattern "ldc ## ldc##" plus a operation instruction. If find such pattern, the method will call method simpleOptimize(). The simpleOptimized() method will first find out the value contained in the constant pool, and pass that into the method called findOptimizedMultivariateInsHandle(), which will find a suitable instruction. And finally delete the old instructions.

2. Variable Folding

   Variable Folding have a different structure compare to Simple Folding. The flow chart illustrates how the method works.



After get the instruction, the method will first find out which category it is. If it is a store instruction, then get the value from the previous instruction and store it into the hashmap call localVar<Integer, Object>. If it is a numeric operation, then find out which kind of operation it is (unary or multivariate). If it is unary, find the nearest 1 value. If it is multivariate, find the nearest two values. If the value is load from Local Variable Slot, then the method findValue() will look up the map and retrieve the value from the map. After that, thing get similar to simpleFolding(). Generate the new instructions and delete the ole instructions.

There are many different instructions, so for each category of operation, there is a method call findOptimizedXXXXXHandle(), which used to determine what exacle the instruction is and generate new InstructionHandle.

Problem with the program

The program cannot handle loops. We try different method but all failed. So in the program, every time it have a method that contains a loop, it will jump over the method and do the optimization for the next method.