

CS614: Advanced Compilers

Fall 2023 (Due: September 30th, 2023)

Assignment A2: Inline me if you can!

1 Assignment Objective

Use JavaCC and JTB to perform a transformation pass that inlines methods based on RTA.

2 Method Inlining

Method Inlining is the process of replacing a method call with the body of the method at a call-site. This optimization eliminates overhead of a method invocation and can bring in various optimization opportunities. In this assignment you are supposed to determine whether a call-site is inlineable based on RTA. If it is inlineable then make appropriate changes to the source code to perform inlining.

3 Detailed Specification

Our task is to iterate over the body of instance methods and perform the following tasks.

- **Determine inlineability:** Determine if the call-site is inlineable based on RTA. Recall that RTA is an improvement over CHA where the call-sites can be more precisely identified using the knowledge of instantiated types.
- **Inlining transformation:** After you have identified the set of `<inlineable-call-site, method>`, perform the following subtasks for performing method inlining at a call-site.
 - Map the method arguments (if any) to the local variables. You can create new assignments or even reuse the existing variables.
 - Replace the method call with the method body. Ensure to maintain syntactic and semantic correctness while doing so.
 - Map the return value to the appropriate local variable. Remember that our grammar supports return only from a single point in a method.

Constraints

- Infinite memory. (No size constraints for inlining)
- Nested inlining is not supported. i.e. The inlined method body will be the original version of that method. Section 7 contains an example describing the same.
- Library calls can be ignored. Only inline methods whose source code is available.
- Input test cases will not contain any form of recursive method calls.
- Input test cases will not contain overloaded methods.
- There is no specification for renaming variables. i.e You can rename variables such that the program follows syntactic and semantic correctness. (Input test cases will not contain variable names with underscore character.)

4 Public Testcase

4.1 Input

```
class Test {
    public static void main(String[] arg) {
        Shape s;
        int ret;
        s = new Circle(); // call to Circle.foo() inherited from Shape.
        ret = s.foo();
        System.out.println(ret);
    }
}

class Shape {
    public int foo() {
        Shape s;
        Circle c;
        Square sq;
        int val;
        int ret;
        s = new Circle();
        val = 10;
        ret = s.computeArea(val); // computeArea() is inlineable since only
                                   Circle is instantiated in the hierarchy.
        return ret;
    }

    public int computeArea(int val) {
        return val;
    }
}

class Circle extends Shape {
    public int computeArea(int diameter) {
        int radius;
        int area;
        int t1;
        int t2;
        int t3;
        t2 = 2;
        t3 = 3;
        radius = diameter / t2;
        t1 = radius * radius;
        area = t1 * t3;
        return area;
    }
}
```

```
class Square extends Shape {  
    public int computeArea(int side) {  
        int area;  
        area = side * side;  
        return area;  
    }  
}
```

4.2 Output

```
class Test {  
    public static void main(String[] arg) {  
        Shape s;  
        int ret;  
        s = new Circle();  
        ret = s.foo();  
        System.out.println(ret);  
    }  
}  
  
class Shape {  
    public int foo() {  
        Shape s;  
        Circle c;  
        Square sq;  
        int val;  
        int ret;  
        int circle_diameter;  
        int circle_radius;  
        int circle_area;  
        int circle_t1;  
        int circle_t2;  
        int circle_t3;  
        s = new Circle();  
        val = 10;  
        circle_diameter = val;  
        circle_t2 = 2;  
        circle_t3 = 3;  
        circle_radius = circle_diameter / circle_t2;  
        circle_t1 = circle_radius * circle_radius;  
        circle_area = circle_t1 * circle_t3;  
        ret = circle_area;  
        return ret;  
    }  
  
    public int computeArea(int val) {  
        return val;  
    }  
}
```

```
class Circle extends Shape {
    public int computeArea(int diameter) {
        int radius;
        int area;
        int t1;
        int t2;
        int t3;
        t2 = 2;
        t3 = 3;
        radius = diameter / t2;
        t1 = radius * radius;
        area = t1 * t3;
        return area;
    }
}

class Square extends Shape {
    public int computeArea(int side) {
        int area;
        area = side * side;
        return area;
    }
}
```

5 Evaluation

Your submission must be named as `rollnum-a2.zip`, where `rollnum` is your roll-number in small letters. Upon unzipping the submission, we should get a directory named `rollnum-a2`. The main class inside this directory should be named `Main.java`. Your program should read from the standard input and print to the standard output. You can leave all the visitors and syntax-tree nodes as it is, but remember to remove all the `.class` files and `.jar` files.

We would run the following commands as part of the automated evaluation process:

- `javac Main.java`
- `java Main < test > out`

If the contents of `out` file match with the expected output for the testcase, you would get marks for the corresponding testcase. Make sure to verify using the CompL Evaluator before submission.

6 Plagiarism Warning

You are allowed to discuss publicly on class and slack, but are supposed to do the assignment completely individually. We would be using sophisticated plagiarism checkers, and if similarity is found, the penalty used in the course would be as follows:

- First instance: 0 marks in the assignment
- Second instance: FR grade.
- Third instance: Report to institute disciplinary committee.

7 Appendix

- Before inlining the call-site `bar(y)`.

```
public class Example {

    public int foo() {
        int x;
        int y;
        y = 10;
        x = bar(y);
        return x;
    }

    public int bar(int a) {
        int b;
        b = foobar(a);
        return b;
    }

    public int foobar(int a) {
        ...
    }

}
```

- After inlining the call-site `bar(y)`.

```
public class Example {

    public int foo() {
        int x;
        int y;
        int bar_a;
        int bar_b;
        y = 10;
        bar_a = y;
        bar_b = foobar(bar_a);
        // foobar is not inlined.
        x = bar_b;
        return x;
    }

    public int bar(int a) {
        int b;
        b = foobar(a);
        return b;
    }

    public int foobar(int a) {
        ...
    }

}
```

-*-*- Do the assignment honestly; enjoy learning the course. -*-*-