
Software Engineering

<http://proglang.informatik.uni-freiburg.de/teaching/swt/2009/>

Exercise Sheet 12

Exercise 1: Tracking Dependencies (15 Points)

Consider the following simple imperative program. The function `read()` reads a number from the console and returns it. The function `write()` writes a number to the console.

```
void main()
{
    int a, b, sum, mul;
    sum = 0;
    mul = 1;
    a = read();
    b = read();
    while (a <= b)
    {
        sum += a;
        mul *= a;
        a++;
    }
    write(sum);
    write(mul);
}
```

Exercise 1.1: Effects of statements

Name for each statement in the above program the set of variables which are *read* and the set of variables which are *written* by the statement.

Exercise 1.2: Control-Flow-Graph

In a Control-Flow-Graph, nodes represent program locations and are labelled with statements. Edges are used to represent jumps. There is an edge from statement A to B iff there is an execution of the program where B executes directly after A.

Draw the Control-Flow-Graph of the above program. Use a dedicated entry node labelled 'Entry: main' and a dedicated exit node labelled 'Exit'.

Exercise 1.3: Control Dependencies

Use the Control-Flow-Graph from Exercise 1.2. Annotate it with control dependencies. There should be a control-dependency edge from node A to node B iff B is control-dependent on A.

Exercise 1.4: Data Dependencies

Use the Control-Flow-Graph from Exercise 1.2. Annotate it with data dependencies. There should be a data-dependency edge from node A to node B iff B is data-dependent on A.

Exercise 2: Keeping a Debugging Logbook (5 Points)

Download the java file `Tree.java` from the homepage. The class `Tree` implements a binary search tree. The function `insert` is supposed to insert a new node into the tree. There are at least two defects in the implementation.

Systematically debug the program and try to find the defects. While doing this, keep a debugging logbook where you explicitly write down all your hypotheses and observations.