

Software Engineering - Exercise Sheet 12

Martin Mehlmann
(`mehlmann@informatik.uni-freiburg.de`)

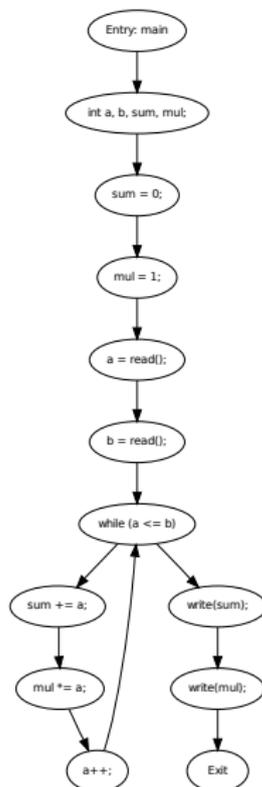
July 23, 2009

Exercise 1.1: Effects of statements

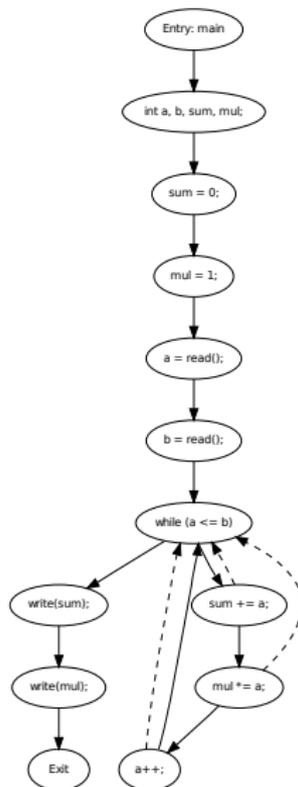
statement	read	write
int a, b, sum, mul;	-	a, b, sum, mul
sum = 0;	-	sum
mul = 1;	-	mul
a = read();	-	a
b = read();	-	b
while (a <= b)	a, b	-
sum += a;	sum, a	sum
mul *= a;	mul, a	mul
a++;	a	a
write(sum);	sum	-
write(mul);	mul	-

Exercise 1.2: Control-Flow-Graph

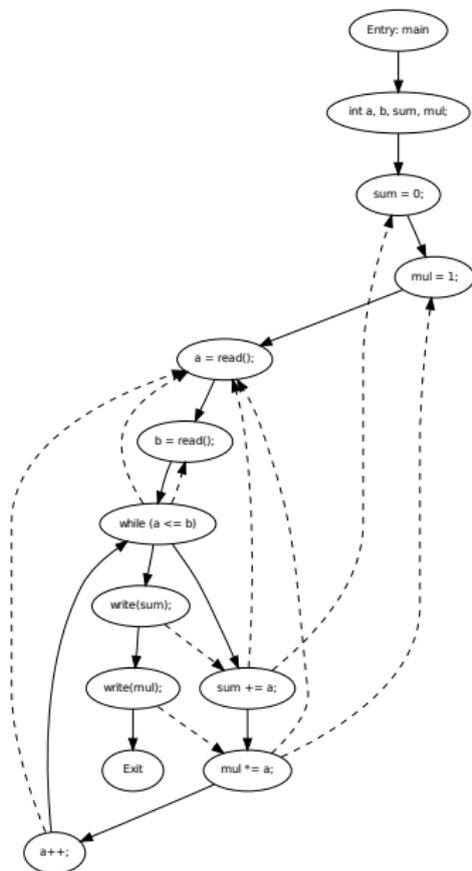
```
void main()  
{  
    int a, b, sum, mul;  
    sum = 0;  
    mul = 1;  
    a = read();  
    b = read();  
    while (a <= b)  
    {  
        sum += a;  
        mul *= a;  
        a++;  
    }  
    write(sum);  
    write(mul);  
}
```



Exercise 1.3: Control Dependencies



Exercise 1.4: Data Dependencies



Exercise 2: Keeping a debugging logbook 1/2

Hypotheses: The program fails if we insert two nodes with the same value.

Test: execute “java Tree x x” for some integer x.

Observation: The program ends up in an endless loop.

Exercise 2: Keeping a debugging logbook 2/2

Hypotheses: If we add another else branch to the if statement in the while loop where we break the while loop iff there is a node with the same value, the program no longer fails.

Test: Modify while loop, recompile and execute “java Tree x x” for some integer x

Modified code:

```
while (next != null) {  
    last = next;  
    if (value < next.data) {  
        next = next.left;  
    } else if (value > next.data) {  
        next = next.right;  
    } else break;  
}
```

Observation: The program no longer ends up in an endless loop but there are two nodes with the same value in the tree.

Exercise 2: Keeping a debugging logbook 2/2

Hypotheses: If we modify the if statement after the while loop such that we do nothing iff there is a node with the same value, the program no longer fails.

Test: Modify program code, recompile and execute “java Tree x x” for some integer x

Modified code:

```
if (value < last.data) {
    last.left = new TreeNode (value);
} else if (value > last.data) {
    last.right = new TreeNode (value);
}
```

Observation: The program no longer fails.