Prof. Dr. Peter Thiemann
Sergio Feo-Arenis
Sergiy Bogomolov

Summer Term 2014

Software Engineering

http://proglang.informatik.uni-freiburg.de/teaching/swt/2014/

Exercise Sheet 5

Exercise 1: Basic Paths (10 Points)

Write down the basic paths for the bubble sort algorithm given below. Before doing that, fill in the missing loop invariants L_1 and L_2. You will need the following auxilary predicates:

1. $\mathsf{sorted}(a, i, j)$ - a is sorted in the range $[i, j]$.
2. $\mathsf{partitioned}(a, i, j, j+1, k)$ - a is partitioned such that each element in the range $[i, j]$ is at most (less then or equal to) each element in the range $[j+1, k]$.

```
@pre ⊤
@post sorted(rv, 0, |rv| − 1)
bool BubbleSort(int[] a_0) {
  int[] a := a_0;
  for
    @L_1 : ???
    (i := |a| − 1; i > 0; i := i − 1) {
    for
      @L_2 : ???
      (j := 0; j < i; j := j + 1) {
      if (a[j] > a[j + 1]) {
        int t := a[j];
        a[j] := a[j + 1];
        a[j + 1] := t;
      }
    }
  }
  return a;
}
```

Exercise 2: Verification Condition Generation (10 Points)

Generate the VCs for the following basic paths and check their validity assuming all the variables are reals:

(1)

```
@ x > 0;
x := x − k;
assume k ≤ 1;
@post x ≥ 0
```

(2)

@ $\top$;
`assume` $k \leq x$;
$x := x - k$;
@`post` $x \geq 0$

(3)

@ $\top$;
$x := x - k$;
`assume` $k \leq x$;
@`post` $x \geq 0$

(4)

@ $k \geq 0$;
$x := x - k$;
`assume` $k \leq x$;
@`post` $x \geq 0$

(5)

@ $y \geq 0$;
$x := x + 1$;
`assume` $x > 0$;
$y := y + x$;
@`post` $x + 2y \geq 3$

Submission

- Submit this sheet *before* the lecture of Thursdays.
- Late submissions will not be accepted.
- Deadline: Thursday 11:59 a.m..