Prof. Bernd Finkbeiner, Ph.D.
Martin Zimmermann, Ph.D.
Leander Tentrup, B.Sc.

Summer term 2013
Problem Set 16
September 30, 2013

Verification

Tool: πVC

You should use the tool πVC to solve the problems of this set. A modified version of the tool for this lecture is available from the lecture website. In order to run it, run the command `java -jar PiGui.jar`. πVC is also installed on our virtual machine, it can be started from the quick lunch bar. You will need an internet connection to successfully compile and verify programs.
Hint 1: The small window on the bottom right might be to small to see all the contents on a small screen. It does not have scoll bars, but it is scrollable to ensure your access to the vital information which properties are proven and which are not.
Hint 2: If you get stuck with the proof you can ask us for some additional hints.

Problem 1: Abs [4 Points]

Prove total correctness of the program Abs using πVC, i.e. annotate the function with an inductive loop invariant and give a ranking annotation. You can download the file `Abs.pi` from the lecture website.

```
@pre ⊤
@post ∀i.0 ≤ i < |rv| → rv[i] ≥ 0
int[] abs(int[] a_0) {
  int[] a := a_0;
  for
    @ ⊤
    (int i := 0; i < |a|; i := i + 1) {
    if (a[i] < 0) {
      a[i] := −a[i];
    }
  }
  return a;
}
```

Figure 1: Computing the absolute values of an array

The following exercises belong to the afternoon session.

Problem 2: Insertion Sort [8 Points]

Prove total correctness of the program InsertionSort using πVC, i.e. annotate the function with inductive loop invariants and give ranking annotations. You can download the file `InsertionSort.pi` from the lecture website.

```
@pre ⊤
@post sorted(rv, 0, |rv| − 1)
int[] InsertionSort(int[] a_0) {
  int[] a := a_0;
  for @ ⊤
    (int i := 1; i < |a|; i := i + 1) {
    int t := a[i];
    for @ ⊤
      (int j := i − 1; j ≥ 0; j := j − 1) {
      if (a[j] ≤ t) break;
      a[j + 1] := a[j];
    }
    a[j + 1] := t;
  }
  return a;
}
```

Figure 2: InsertionSort