FoVeOOS competition 2011
First Challenge

Krakatoa/Why3 team

October 4, 2011

1 Solution in Why3

The Why3ML version of the challenge code, is as follows. (verbatim copy, sorry for the length file).

```
(*
COST Verification Competition
Please send solutions to vladimir@cost-ic0701.org

Challenge 1: Maximum in an array


Given: A non-empty integer array a.

Verify that the index returned by the method max() given below points to
an element maximal in the array.
*)


module Max

  use import int.Int
  use import int.MinMax
  use import module ref.Ref
  use import module array.Array

  let max (a: array int) =
    { 0 < length a }
    let x = ref 0 in
    let y = ref (length a - 1) in
    while !x <> !y do
      invariant { 0 <= !x <= !y < length a /\
          forall i:int.
```

```
            0 <= i < !x \/ !y < i < length a ->
                a[i] <= max a[!x] a[!y]
                }
      variant { !y - !x }
      if a[!x] <= a[!y] then x := !x+1 else y := !y-1
    done;
    !x
    { 0 <= result < length a /\
      forall i:int. 0 <= i < a.length -> a[i] <= a[result] }

end

(*
Local Variables:
compile-command: "why3ide challenge1_why3.mlw"
End:
*)
```

The post-condition specifies the expected functional behavior of the code: forall index i (within the bounds of the given array of course) the value at this index is smaller or equal to the one at the index returned.

To perform the proof, a loop invariant is needed: we just specify that all the values outside the interval $x..y$ are smaller or equal to the largest of $a[x]$ and $a[y]$.

Termination is also guaranteed thanks to the given loop variant.

In Why3, a single formula is produced, which is that the precondition implies the weakest precondition of the post. It is proved automatically with several automatic provers.

Proof obligations	Alt-Ergo 0.93	CVC3 2.2	Yices 1.0.25	Z3 2.19
parameter max	3.40	0.06	0.06	0.04

Note: this table is automatically produced by the "why3replayer" tool.

2 Solution in Java

The following file is the Java version of the same code, annotated in the Krakatoa-specific variant of JML. Most of the annotations are similar to those in Why3. Notice that the logic function max is not known by Krakatoa so it has to be given and axiomatized.

```
/*
COST Verification Competition
Please send solutions to vladimir@cost-ic0701.org

Challenge 1: Maximum in an array


Given: A non-empty integer array a.

Verify that the index returned by the method max() given below points to
an element maximal in the array.

*/

/*@ axiomatic integer_max {
  @   logic integer max(integer x, integer y);
  @   axiom max_is_ge : \forall integer x y; max(x,y) >= x && max(x,y) >= y;
  @   axiom max_is_some : \forall integer x y; max(x,y) == x || max(x,y) == y;
  @ }
  @*/

public class Max {


    /*@ requires a.length > 0;
      @ ensures 0 <= \result < a.length &&
      @   \forall integer i; 0 <= i < a.length ==> a[i] <= a[\result];
      @*/
    public static int max(int[] a) {
        int x = 0;
        int y = a.length-1;
        /*@ loop_invariant 0 <= x <= y < a.length &&
          @      \forall integer i;
          @         0 <= i < x || y < i < a.length ==>
          @         a[i] <= max(a[x],a[y]);
          @ loop_variant y - x;
          @*/
        while (x != y) {
            if (a[x] <= a[y]) x++;
                else y--;
        }
        return x;
    }


}
```

```
/*
Local Variables:
compile-command: "make challenge1.why3ml"
End:
*/
```

The proof proceeds automatically by several theorem provers.

Proof obligations	Alt-Ergo 0.93	CVC3 2.2	Vampire 0.6	Yices 1.0.25	Z3 2.19
Method max, default behavior		3.25		0.12	2.20
index bounds	0.08			0.11	0.08
Constructor of class Max, default behavior	0.04		0.02	0.06	0.02
Constructor of class Max, Safety	0.06		0.02	0.06	0.02