15-820-a
Assignment 5
Verification of ANSI-C with PVS

Due Apr. 30, 2003

1 Find the Minimum

1. Write a function in ANSI–C that finds the minumum number in an array. The size of the array is passed as a parameter.
2. Translate your ANSI–C code into PVS language, as described in the class. You may assume that the ANSI–C integer type is unbounded, i.e., matches the PVS type `integer`.
3. Write a specification in PVS language as a theorem.
4. Informally state the invariant needed for the proof of correctness of this theorem. You can do this as a comment in the PVS file you hand in.
5. Formally write this invariant as a PVS theorem.
6. Prove that the invariant implies your correctness claim using the PVS theorem prover.
7. Prove your invariant! You will probably need help for this. If you get stuck, put the files you are working on in a directory on AFS. Make this directory read and writeable to `kroening` and call 85409 or write to `kroening@cs.cmu.edu`.

2 Binary Search

The function with the prototype

```
bool binary_search(unsigned int size,
                   const int a[],
                   int x);
```

is supposed to return true if and only if the value x is to be found in the array a. The array a is assumed to have size elements. The array is assumed to be sorted in ascending order.

1. Write the body of the function using a loop (no recursion).
2. Translate your ANSI–C code into PVS language, as described in the class. You may assume that the ANSI–C integer type is unbounded, i.e., matches the PVS type integer.
3. Formalize ”sorted in ascending order” as a PVS predicate.
4. Write a specification in PVS language as a theorem.
5. Informally state the invariant needed for the proof of correctness of this theorem. You can do this as a comment in the PVS file you hand in.
6. Formally write this invariant as a PVS theorem.
7. Prove that the invariant implies your correctness claim using the PVS theorem prover.
8. Prove your invariant!