Interactive Software Verification SS 2013

https://www21.in.tum.de/teaching/isv/SS13

H. Gast, L. Noschinski

SHEET 5

First proofs with SimplC

Date: 5.06.2013
Hand-In: 12.16.2013, 8:30

Goals Perform first correctness proofs and gain experience with the concepts from the lecture.

Getting Started Download `SimplC.zip` from the website and follow the README file.

Exercise 1 [2] Simple Proofs

Formulate and prove the following properties:

1. `j = i + i + i;` results in `j = 3*i`
2. `i = i + 1; i = i - 1;` does not change `i` w.r.t. start state
3. `if (i != 0) i = 0;` always results in `i = 0`.
4. `j = a[i];` results in aget a i being the value of `j`, if appropriate preconditions are met

Exercise 2 [4] Empty arrays

We define the following concepts: An array is *empty*, if all entries are 0. We can interpret an `int` as a boolean value.

is-empty a $\equiv$ ($\forall$ i $\in$ {0..<alength a}. aget a i = 0)
to-bool (i :: int) $\equiv$ i $\neq$ 0

Prove now that the following program checks whether an array is empty, i.e., whether the postcondition `to-bool r = is-empty a` is fulfilled. You will need to invent an appropriate precondition and invariant.

```
i = 0;
r = 1;
while (i != n && r != 0) {
    if (a[i] != 0) {
        r = 0;
    }
    i = i + 1;
}
```

Hints

- Unfold the the definitions of is-empty and to-bool (with `unfolding`).
- On goals with a lot of boolean combinators, use the method `split_vc` to get an overview over the verification conditions.
- A witness for goals of the form $\exists$k$\in$A. ... you can use the following command to choose a witness:
  apply (rule-tac x="..." in bexI)

Exercise 3 [4] Sorted arrays

We say an array is *sorted*, if it fulfills the following predicate:

sorted-arr a $\equiv$ $\forall$ i $\in$ {0..<alength a - 1}. aget a i $\leq$ aget a (i+1)"

Prove that the following programm decides whether an array is sorted. You will need to invent an appropriate precondition and invariant.

```
i = 0;
r = 1;
while (i < n - 1) {
    if (a[i+1] < a[i]) {
        r = 0;
        break;
    }
    i = i + 1;
}
```