Assignment 2

Proving loop invariants

Static Program Analysis and Constraint Solving
Master's Degree in Formal Methods in Computer Science
Year 2021/22

Submission deadline: October, 8th

Submission instructions: Students are required to submit a single file with the SMT-LIB script. They might have to present and defend their work after submission.

The following algorithm performs a linear search for an element `x` in an array `a`, assuming that the input array actually contains the target element `x`.

Precondition: $\exists j. 0 \leq j < len(a) \wedge a[j] = x$

```
int linearSearch(int x, int[] a) {
    int i = 0;
    while (a[i] != x) {
      i = i + 1;
    }
    return i;
}
```

Postcondition: $0 \leq res < len(a) \wedge a[res] = x \wedge \forall j. 0 \leq j < res \Rightarrow a[j] \neq x$

In the postcondition, *res* denotes the value returned by the function. The loop maintains the following invariant:

$$0 \leq i \leq len(a) \wedge (\exists j. i \leq j < len(a) \wedge a[j] = x) \wedge (\forall j. 0 \leq j < i \Rightarrow a[j] \neq x)$$

The first conjunct constraints the variable i. The second one specifies that the element being searched for must be in the segment of the array starting from its i-th position. The third one states that the part of the array traversed so far does not contain the element x.

In order to prove this program correct, a program verifier would generate some conditions and prove them correct. Among these conditions, there is one ensuring that the invariant is preserved after every loop iteration. Your task is to use Z3 to prove this condition, which is formalized as follows:

- **Assuming that:**
  - The invariant holds before executing the loop body:
    $0 \leq i \leq len(a) \wedge (\exists j. i \leq j < len(a) \wedge a[j] = x) \wedge (\forall j. 0 \leq j < i \Rightarrow a[j] \neq x)$
  - The `while` condition holds:
    $a[i] \neq x$

  - We denote by i' the value of i after the execution of the loop body:
    $i' = i + 1$

- **Then** the invariant holds after executing the loop body:
  $0 \leq i' \leq len(a) \wedge (\exists j.\, i' \leq j < len(a) \wedge a[j] = x) \wedge (\forall j.\, 0 \leq j < i' \Rightarrow a[j] \neq x)$

In this case, *len* is an uninterpreted function. Write an SMT-LIB script to check that whenever the three assumptions shown above hold, so does the invariant after the execution of the loop body.