Assignment 1 CSSE3100/7100 Reasoning about Programs

Due: 1pm on 5 April, 2022

The aim of this assignment is to consolidate your understanding of the course's material on weakest precondition reasoning. It is worth 20% of your final mark for the course.

Instructions: Submit a single plain text file (preferable) or pdf file with your solution to the following questions to Blackboard by the due date and time. Do not embed images (jpeg, etc.) of your solution if submitting a pdf.

1. Determine whether the program Main is correct using weakest precondition reasoning. Any quantifiers in your reasoning must be eliminated using the one-point rule. The challenge is to put the weakest precondition into a form in which this rule can be applied.

If the program is not correct, provide the weakest requires clause which will make it correct.

```
method Main(u: int) returns (t: int)
    ensures t > u
{
    t := Abs(7 * u)
}
```

where

```
method Abs(x: int) returns (y: int)
    ensures y >= 0 && (y == x || y == -x)
```

(6 marks)

2. You have written the following program Exp to calculate a^n for a positive number n.

```
method Exp(a: int, n: nat) returns (z: int)
{
    z := 1;
    var i := n;
    while (i > 0) {
        z := z * a;
        i := i - 1;
    }
}
```

Your colleague, Klaus, says it can be written more efficiently by decreasing i by more than 1 on particular iterations. He suggests the following program with loop invariant $z * a^i == a^n$ && i >= 0.

```
method ExpK(a: int, n: nat) returns (z: int) {
    z := 1;
    var i := n;
    while (i > 0) {
        if i % 2 != 0 {
            z, i := z * a, i - 1;
        } else {
            z, i := z * z, i / 2;
        }
    }
```

Your other colleague, Erika, says that Klaus's idea is good, but he hasn't got the program quite right. She suggests the following program with loop invariant z * b^i == a^n.

```
method ExpE(a: int, n: nat) returns (z: int) {
    z := 1;
    var i, b := n, a;
    while (i != 0) {
        if i % 2 != 0 {
            z, i := z * b, i - 1;
        } else {
            b, i := b * b, i / 2;
        }
    }
}
```

(a) Use weakest precondition reasoning to show that Klaus's program is incorrect, and that Erika's is partially correct. For Klaus's program, you need to carry out the reasoning to the point where the proof fails, and provide a single counter-example to show that it does fail. You may use the notation a^n in your proofs and must state any mathematical rules for exponents that you use, such as a^0 == 1. **(10 marks)**

(b) Find a termination metric for Erika's program and show total correctness using weakest precondition reasoning. You may strengthen the loop invariant if needed. For this proof indicate the predicates from the proof of part (a) with "..." (as was done in the lecture slides). **(4 marks)**

Marking

You will get marks for

- the application of the appropriate weakest precondition rules for each line of code (do not skip the application of a rule), and
- for correct and, where necessary, justified simplifications.

Simplifications can be justified (where appropriate) using rules from Appendix A of *Programming from Specification* (available on Blackboard). Be sure to mention any steps where you strengthen a predicate.

Marks will also be given for

- justifying why a program is correct or incorrect,
- providing the weakest requires clause, if necessary, in Q1.

School Policy on Student Misconduct

This assignment is to be completed individually. You are required to read and understand the School Statement on Misconduct, available on the School's website at: http://www.itee.uq.edu.au/itee-student-misconduct-including-plagiarism