McMaster University
Department of Computing and Software
Musa Al-hassy

COMP SCI 3EA3
Quiz 3

COMP SCI 3EA3 — Software Specification and Correctness

Feburary 13, 2017

Name _____ Student Number _____

Special Instructions:

- This examination paper includes 5 pages (including this cover page) and 4 questions.
  You are responsible for ensuring that your copy of the paper is complete. Bring any discrepancy to the attention of your invigilator.
- **Read each question completely and carefully** before answering it.
- Answer all questions.
- **In doubt, document!**

Contents

1 When You're Low, You Can Always Look Up — 10 marks — **2**

2 Assign Me To The Moon — 30 marks — **3**

3 (Bonus) Lattice-jutsu — 6 marks — **5**

4 (Bonus) ACSL: "Never Gonna Give You Up!" — 10 marks — **5**

In this quiz, you may use the following properties for question 2.

(0) Precondition:	$0 \leq N$
(1) Successors are strictly above:	$x < x + 1$ and $x - 1 < x$
(2) Order Properties:	$x \leq y < x \equiv \mathit{false}$
(3) Subtraction Distributivity:	$x - (y + z) = (x - y) - z$
(4) Monotonicity of Subtraction in the first argument:	$x - z \leq y - z \Leftarrow x \leq y$
(5) Definition of strict order:	$x \leq y \wedge x \neq y \equiv x < y$
(6) <-arithmetic:	$x + 1 \leq y \equiv x < y$

1 When You're Low, You Can Always Look Up — 10 marks —

Fill in the blank boxes with the appropriate *names* of theorems or properties in the following proof of "lower adjoints can always look up".

Assuming,

$$\text{Galois Connection:} \qquad \forall x, y \bullet L\ x \sqsubseteq y \quad \equiv \quad x \sqsubseteq U\ y$$

we obtain

$$L\,(\sqcup x \mid R \bullet P) = (\sqcup x \mid R \bullet L\ P)$$

using the "principle of indirect equality from above": Let 'a' be arbitrary, and then calculate

$$\begin{array}{ll} & L\,(\sqcup x \mid R \bullet P) \sqsubseteq a \\ = & \quad \{ \quad \boxed{} \quad \} \\ & (\sqcup x \mid R \bullet P) \sqsubseteq U\ a \\ = & \quad \{ \quad \boxed{} \quad \} \\ & (\forall x \mid R \bullet P \sqsubseteq U\ a) \\ = & \quad \{ \quad \boxed{} \quad \} \\ & (\forall x \mid R \bullet L\ P \sqsubseteq a) \\ = & \quad \{ \quad \boxed{} \quad \} \\ & (\sqcup x \mid R \bullet L\ P) \sqsubseteq a \end{array}$$

2 Assign Me To The Moon — 30 marks —

Recall in class that we said a finite quantification $(\oplus i \mid 0 \le i < N \bullet f\ i)$ can be computed with a simple for-loop

```
total = e ;   // e is the unit of oplus
for(int n = 0 ; n < N ; n ++)
  total = oplus( total , f(n) );
```

$$total, n := e, 0 \;;\; \textbf{do}\ n \neq N \to total, n := total \oplus f\ n, n+1\ \textbf{od}$$

with invariant "`total` holds the quantification *so far*",

$$P: \qquad total = (\oplus i \mid 0 \le i < n \bullet f\ i) \quad \wedge \quad 0 \le n \le N$$

Let us prove that it is indeed an invariant and our program is correct. Parts 1 and 3 are the most difficult and so are fill-in-the-blanks. You will need to use some of the properties provided on the first page.

1. (5 marks) The invariant holds initially: $P[total, n := e, 0]$ is true.

   $P[total, n := e, 0]$

   $=$ { definition of P and textual substitution }

   $\boxed{}$

   $=$ { Order Properties —first page of this quiz! }

   $e = (\oplus i \mid false \bullet f\ i) \quad \wedge \quad \boxed{}$

   $=$ { $\boxed{}$ }

   $e = e \quad \wedge \quad \boxed{}$

   $=$ { Reflexitivity of '=' and identity of conjunction }

   $0 \le 0 \le N$

   $=$ { $\boxed{}$ }

   $0 \le N$

   $=$ { $\boxed{}$ }

   $true$

2. (10 marks) After the loop terminates, we have solved our problem:

   $$\neg(n \neq N) \wedge P \quad \Rightarrow \quad total = (\oplus i \mid 0 \le i < N \bullet f\ i)$$

   Hint: start with the complicated part and weaken to obtain the simpler part!

3. (5 marks) The loop body maintains its truthiness: if $P \wedge n \neq N$ then $P[total, n := total \oplus f\ n, n+1]$ is true.
   Assuming $P \wedge n \neq N$, we calculate

$P[total, n := total \oplus f\ n, n+1]$

$=$ { definition of P and textual substitution }

[ ]

$=$ { [ ] }

$total \oplus f\ n = (\oplus i \mid 0 \leq i < n \bullet f\ i) \oplus f\ n \quad \wedge$ [ ]

$=$ { [ ] }

$total \oplus f\ n = total \oplus f\ n \quad \wedge$ [ ]

$=$ { Reflexitivity of '=' and and identity of conjunction }

$0 \leq n+1 \leq N$

$\Leftarrow$ { [ ] }

$0 \leq n \leq n+1 \leq N$

$=$ { [ ] }

$0 \leq n < N$

$=$ { [ ] }

$0 \leq n \leq N \wedge n \neq N$

$=$ { [ ] }

$true$

4. (10 marks) A bound on the number of loop-steps is '$N - n$': it is clearly non-negative before the loop, so we need only prove it gets smaller with each step; that is, for arbitray s,

$$\text{if } P \wedge N - n = s \wedge N \neq n \quad \text{ then } \quad (N-n)[total, n := total \oplus f\ n, n+1] < s.$$

3 (Bonus) Lattice-jutsu — 6 marks —

Prove the following property in the calculational style,

$$a \sqsubseteq a \sqcap b \quad \equiv \quad a \sqsubseteq b$$

Hint: start with the complicated side and simplify.

4 (Bonus) ACSL: "Never Gonna Give You Up!" — 10 marks —

Provide appropriate —as usual, strongest `ensures` and weakest `requires`— specifications for the preamble of the program `ferreira`. (As usual, your answer must be ACSL acceptable; if you wish to use a non-primitive concept, then you must axiomatise it yourself.)

```
/*@
@ requires
@ assigns
@ ensures
*/
int ferreira(unsigned int x, unsigned int y)
{
  return (y <= x) ? ferreira(x - y, y) + 1 : 0;
}
```