McMaster University
Department of Computing and Software
Musa Al-hassy

COMP SCI 3EA3 — Software Specification and Correctness

February 17, 2017

Exercise 7.0 The "Assignment Rule" *Rules!*

Simplify the following —as succinctly as possible—, assuming all subscripts are in range.

1. wp "$i, j := i + 1, j + i$" $(i = j)$
2. wp "$i := i + 1;\ j := j + i$" $(i = j)$
3. wp "$z, x, y := 1, c, d$" $(z * x^y = c^d)$
4. wp "$x := b[i]$" $(x = b[i])$
5. wp"$x := E$" R assuming x does not occur freely in R.
6. wp "$b[i] := 5$" $(b[i] = 5)$
7. wp "$b[i] := 5$" $(b[i] = b[j])$
8. wp "$b[b[i]] := i$" $(b[i] = i)$
9. wp "$b[n] := b[n-1] \oplus b[n]$" $(\ b[n] = (\oplus j \mid 3 \le j < n \bullet b[j])\ )$

Exercise 7.1 Calculating Assignments

Solve for x in the following assignments; do not be 'ad hoc' —guess and check—; instead calculate!
That is, use the definition of wp, or "the assignment rule", to *derive* the assignments —consequently, they will be "correct by construction".

Unknowns may depend on all variables!

1. $\{\ i = j\}\ i, j := i + 1, x \{i = j\}$
2. $\{\ i = j\}\ j := x;\ i := i + 1\ \{i = j\}$
3. $\{\ z + a * b = c\}\ z, a := z + b, x\ \{z + a * b = c\}$
4. $\{\ \text{even}\ a \wedge z + a * b = c\}\ a := a \div 2;\ b := x\ \{z + a * b = c\}$
5. $\{true\}\ a, b := a + 1, x\ \{b = a + 1\}$
6. $\{true\}\ a := a + 1; b := x\ \{a = b\}$ (Hint: why is $x = a + 1$ wrong?)
7. $\{i + p = c\}\ i, p := m + 1, x\ \{i + p = c\}$
8. $\{true\}\ n, total := 0, x\ \{total = (\oplus j \mid 0 \le j < n \bullet b[j])\}$
9. $\{n > 0 \wedge total = (\oplus j \mid 0 \le j < n \bullet b[j])\}\ n, total := n + 1, x\ \{total = (\oplus j \mid 0 \le j < n \bullet b[j])\}$

Exercise 7.2 Swap It!

Prove that the following are swapping algorithms, and do so in 2 ways: as a usual calculation, and as a *proof outline* via bottom-up approach.

1. `t := x ; x := y ; y := t`

   Hint: you want to prove, for arbitrary X and Y,

   $$\mathsf{wp}\ \text{“}t := x; x := y; y := t\text{”}\ (y = X \wedge x = Y) \quad \equiv \quad x = X \wedge y = Y$$

2. `x, y := y, x`
3. `x := x + y ; y := x - y ; x := x - y`
4. `x := x * y ; y := x / y ; x := x / y`
5. `x := x ^ y ; y := x ^ y ; x := x ^ y` where `^` is bit-wise xor.

For the last three: what happens if the values are really big? What if one of the values is 0? What if $x = y$?
Generalise the last three so that we can perform a swap using a pair of inverse functions.

Exercise 7.3 Getting Comfortable With Choice / Selection / Alternatives

1. Define a procedure *fig* so that $fig(x, n)$ prints either squares or triangles of size $n \times n$ according to whether x is ‘S’ or ‘T’. Implement this method in `C` using the GCL notation defined in `alhassy_gcl2.c`.
2. Given your answer to the previous exercise, what is printed by `fig('H',-1)`? [Hint: in at least one of these cases the procedure will probably be equivalent to a sequence which contains a meaningless instruction.]
3. Using the definition, prove the order of arms in a selection does not matter, for the case of two arms

   $$\textbf{if}\ B_1 \to S_1 \ [] \ B_2 \to S_2\ \textbf{fi} \quad \approx \quad \textbf{if}\ B_2 \to S_2 \ [] \ B_1 \to S_1\ \textbf{fi}$$

4. Use `alhassy_gcl2.c` —which implements guarded commands non-deterministically— to write a program in `C` to “increase x non-deterministically by an arbitrary amount”.

   Discuss why such a program is “bad” and suggest a possible solution.

Exercise 7.4 — Programming Project: Optional Assignment, 5% —due before the next quiz

Do the previous “Optional Assignment”, at its original worth,

OR

for 5%, write a correct version of the algorithm on page 30, Exercises 3.6, of the course text, that given four numbers a, b, c, d will present a proof derivation solving for X in $\sqrt{a} + \sqrt{b}\ X\ \sqrt{c} + \sqrt{d}$ as is done on pages 28-29.
Write and submit a report similar to the indications on Sheet 5. —including, Frama-C annotations, troubles encountered, and possible future directions—