CSC410, Fall 2016 - Homework 2

Name: Siyuan Zheng

Student Number: 1000726814

Lecture: Tuesday

I am the solo author of this homework, signature: Siyuan Zheng

/* Note: Due to compilation between word and pdf, '¥' in this pdf file refers to backslash */

Problem 1

(a) No. Since (i >= 0 && i < arr.length) does not necessarily imply i < arr.length/2

Postcondition:

(¥forall int i; (i >= 0 && i < arr.length/2) =>(arr[i] == ¥old(arr[i] - 1))

(b) No. Since arr[] has certain length, arr[non-valid index] cannot be evaluated to be compared with x.

Postcondition:

!¥result ⇔ (¥forall int i; i>=0 && i < arr.length; arr[i] != x)

Problem 2

(a) int arr = [4];

(b) x = -5

Problem 3

(a) Program: if z > 20 then y = y + 5z + 410 else y = y + 410

Q: y >410

wp(Program, Q)

=wp(if z > 20 then y = y + 5z + 410 else y = y + 410 , y > 410)

$=(z > 20 \Rightarrow wp(y = y + 5z + 410,\ y > 410) \wedge (z <= 20 \Rightarrow wp(y = y + 410,\ y > 410))$

$wp(y = y + 5z + 410,\ y > 410)$

$= y + 5z + 410 > 410$

$= y + 5z > 0$

$wp(y = y + 410,\ y > 410)$

$= y+410>410$

$= y > 0$

$wp(\text{if } z > 0 \text{ then } y = y + 5z + 410 \text{ else } y = y + 410,\ y > 410)$

$=(z > 20 \Rightarrow y + 5z>0) \wedge (z <= 20 \Rightarrow y > 0)$

$=(z <= 20 \vee y+5z>0) \wedge (z > 20 \vee y > 0)$

Hoare triple method:

$y > 410[y \leftarrow y + w,\ w \leftarrow 410]\ \{y := y + w,\ w := 410\}\ y > 410$ (axiom of assignment)

$y + w > 410\ [w \leftarrow 410]\ \{y := y + w,\ w := 410\}\ y > 410$

(by the definition of substitution algorithm)

$y + 410 > 410\ \{y := y + w,\ w := 410\}\ y > 410$ (by definition of substitution algorithm)

$y > 0\ \{y := y + w,\ w := 410\}\ y > 410$ (by some axiom of arithmetic)

$y > 0 \wedge z <= 20\ \{y := y + w,\ w := 410\}\ y > 410$ (rule of consequence 2)

(since $y > 0 \wedge z <= 20 \Rightarrow y > 0$)

$y > 410[y \leftarrow y + w,\ w \leftarrow 410 + 5z]\{y := y + w,\ w := 410 + 5z\}y > 410$

(axiom of assignment)

$y + w > 410[w \leftarrow 410 + 5z]\ \{y := y + w,\ w := 410 + 5z\}y > 410$

(by the definition of substitution algorithm)

$y + 410 + 5z > 410\ \{y := y + w,\ w := 410 + 5z\}y > 410$

(by the definition of the substitution algorithm)

$y + 5z > 0\ \{y := y + w,\ w := 410 + 5z\}y > 410$

(by some axioms of arithmetic)

$y + 5z > 0 \land z > 20\{y := y + w,\ w := 410 + 5z\}y > 410$
(rule of consequence 2)

Then we have $y > 0 \land z <= 20\ \{y := y + w,\ w := 410\}\ y > 410$ AND $y + 5z > 0 \land z > 20\{y := y + w,\ w := 410 + 5z\}y > 410$

Compared to the weakest precondition before, which is wp(Program,Q)
$=(z > 20 \Rightarrow y + 5z>0) \land (z <= 20 \Rightarrow y > 0)$
Both Hoare tripe method and weakest precondition method yield the same result.

(b)Weakest precondition method:
wp(if B then S1, P)
$=(B \Rightarrow wp(S1,P) \land (\neg B \Rightarrow P)$
wp(if B1 S2 else S3, P)
$=B1 \Rightarrow wp(S2,P) \land (\neg B \Rightarrow wp(S3,P))$
Here B is the first if statement, S1, is the outer if branch
B1 is the sub if statement, S2 is the sub if branch
S3 is the else branch for B1

wp(if x > y then S1, z > 410)
$=(x > y \Rightarrow \underbrace{wp(S1,z>410)}_{①}) \land (x <= y \Rightarrow z > 410)$

①wp(S1,Z>410)
Here S1= x=x -10, if B1 then S2 else S3

Then wp = (S1, z > 240)
$=(x-10< y-10 \Rightarrow \underbrace{wp(z += 10,\ z > 410)}_{②}) \land (x-10>= y -10 \Rightarrow \underbrace{wp(z -= 10,\ z > 410)}_{③})$

②wp(z += 10, z > 410)

= z + 10 > 410

= z > 400

= wp(z += 10, z > 410)

= z > 400

③wp(z -= 10, z > 410)

= z -10>410

= z > 420

= wp(z += 10, z > 40)

= z > 420

Then, wp(if x > y then S1, z > 240)

= wp(Program, z > 240)

= (x > y => (x < y => z > 400) ∧ (x >= y => z > 420))

∧(x <= y => z > 410)

= (x > y => z > 420) ∧(x <= y => z > 410)

Thus , wp(Program, z > 410)

= (x > y => z > 420) ∧ (x <= y => z > 410)

Hoare triple method:

z > 410 [z ← z - 10]{z := z -10} z > 410

(axiom of assignment)

z - 10 > 410 {z := z -10}z > 410

(by the definition of the substitution algorithm)

z > 420 {z := z -10} z > 410

(by some axioms of arithmetic)

(x =10 >= y -10) ∧ z > 420 {z := z -10}z > 410

(by rule of consequence 2)

(1), $x >= y \wedge z > 420\{z := z - 10\}\ z > 410$

(by some axioms of arithmetic)

$z > 410[z \leftarrow z + 10]\{z := z + 10\}z > 410$

(axiom of assignment)

$z + 10 > 410\{z := z + 10\}z > 410$

(by the definition of the substitution algorithm)

$z > 400\ \{z := z + 10\}z > 410$

(by some axiom of arithmetic)

(2), $z > 400 \wedge x < y\ \ \{z := z + 10\}\ z > 410$

(by rule of consequence 2)

we have (1), (2)

$x >= y\ \wedge\ z > 420[z \leftarrow 415]\{z := 415\}\ x >= y\ \wedge\ z > 420$

(by axiom of assignment)

$x >= y\ \wedge\ 415 > 420[z := 415]\ x >= y\ \wedge\ z > 420$

(by the definition of substitution algorithm)

(3)$False[z := 415]\ x >= y\ \wedge\ z > 420$

$z > 400 \wedge x < y\ [z \leftarrow 415]\{z := 415\}\ z > 400 \wedge x < y$

(by axiom of assignment)

$415 > 400\ \wedge x < y\ \{z := 415\}\ z > 400 \wedge x < y$

(by the definition of substitution algorithm)

(4) $x < y\ \{z := 415\}\ z > 400 \wedge x < y$

(by some axiom of arithmetic)

When combining (2) and (4),

We get

(5), $X < y\{z ;= 415,\ z := z + 10\}\ z > 410$

And $X < y\{z ;= 415,\ z := z + 10\}\ z > 410$

$\Rightarrow (x > y \Rightarrow z > 420) \wedge (x <= y \Rightarrow z > 410)$

Then (5) => wp(Program, z > 410),

Which is appropriate

Problem 4

(a)

WTS $I \wedge B \{S\} I$

$i <= n\ [x \leftarrow x * y\ ,\ i \leftarrow i + 1]\{\ x := x * y\ ,\ i := i + 1\}\ i <= n$

(by axiom of assignment)

$i + 1 <= n\ \{\ x := x * y\ ,\ i := i + 1\}\ i <= n$

(by the definition of the substitution algorithm)

$i <= n - 1\ \{\ x := x * y\ ,\ i := i + 1\}\ i <= n$

(by some axioms of arithmetic)

$i < n\ \{\ x := x * y\ ,\ i := i + 1\}\ i <= n$

(by some axioms of arithmetic)

$i < n \wedge i < n\ \ \{\ x := x * y\ ,\ i := i + 1\}\ i <= n$

Then i <= n-1 is a loop invariant,

Now it is obvious that when i = n -1, the loop is still running, and in the loop code, i get incremented, I now equals to n - 1 + 1 = n, and i is only about to finish the loop segment. Thus i <= n is a loop invariant

(b) $x == pow(y,i)$

$x == pow(y,i)[x \leftarrow x * y,\ i \leftarrow i + 1]\{x = x * y,\ i = i + 1\}x == pow(y,\ i)$

(By axiom of assignment)

$x * y == pow(y,i+1)\ \{x = x * y,\ i = i + 1\}x == pow(y,\ i)$

(by the definition of substitution algorithm)

Since $x == pow(y,i) \Rightarrow x * y == pow(y,y+1)$

Then x == pow(y, i){x = x * y, i = i + 1}x == pow(y, i)

(By the rule of consequence 2)

Then I ∧ B{S}I, x == pow(y, i) is a loop invariant.

Problem 5

(a) I don't have time as well as I don't know how to answer this question.

(b) I don't have time as well as I don't know how to answer this question.