CSCI 2600 — Principles of Software

# Jimmy Chen - Homework 1

## Problems

### Problem 1 (12 pts., 2 pts. each): Condition Strength

Order conditions from **the strongest to the weakest** by showing the longest implication chain of the form A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D ... (where A, B, C, D, ... are given conditions, $\rightarrow$ is implication). If total ordering by strength cannot be established, show all partial orderings. Prove by showing all implications. If any implication is false, provide a counterexample. You must show all work in order to get full credit. Unless otherwise stated, assume all referenced variables are defined as integers.

**(1)** A = { x = 2k + 1 $\wedge$ x = x + 1 $\wedge$ y = 10 }
B = { x is divisible by 6 $\wedge$ y = x + 2 }
C = { x = 18 $\wedge$ y = 20 }
D = { x % 2 = 0 $\wedge$ y is even }

Implication Chain: $(C) \rightarrow (D) \rightarrow (B) \rightarrow (N/A)$
Proof of implication: $A \rightarrow B$ : A isn't in the ordering since it's a invalid statement. We have x = 18, and y = 20, so x can't be sum 2k+1. But for the rest, since x is divisible by 6, then x is even, and y = x + 2, so y is even. So B is stronger than D. And D is stronger than C.

**(2)** A = { 5 $\leq$ k < 5 }
B = { k $\geq$ -10 }
C = { -10 < k $\leq$ 1 }
D = { 10 $\leq$ k $\leq$ -10 }

Implication Chain: $(B) \rightarrow (C) \rightarrow (N/A) \rightarrow (N/A)$
Proof of implication: $A \rightarrow B$ : A and D isn't in the ordering since it's a invalid statement. We have k $\geq$ -10, so B is stronger than C as it implies that k is between -10 and 1.

**(3)** A = { x $\geq$ 0 $\wedge$ y > x }
B = { x = 3 $\wedge$ y > 10 }
C = { x = y % 10 }
D = { xy = 0 }
E = { x > 1 $\vee$ y > 1 }

Implication Chain: $(A) \rightarrow (B) \rightarrow (N/A) \rightarrow (N/A) \rightarrow (N/A)$
Proof of implication: $A \rightarrow B$ : True ...

**(4)** Recall the names for infinite sets of numbers used in math (e.g., see "Peter Jephson Cameron (1998). *Introduction to Algebra*. Oxford University Press. p. 4. ISBN 978-0-19-850195-4.")
A = { z $\in \mathbb{Q}$ }
B = { z = $\sqrt{-1}$ }
C = { z $\in \mathbb{N}$ }
D = { z = m / n }
E = { z $\in \mathbb{Z}$ }
F = { z = y % 2 }
G = { z $\in \mathbb{R}$ }
H = { z = y / 2 }

Implication Chain: $(C) \to (E) \to (A) \to (D) \to (G)$
Proof of implication: $A \to B$ : This implication is false because z can't be mod 2 as it includes only 0 and 1, which doesn't follow the logic chain. Z being a quotient of two numbers doesn't imply that z is a real number. So the implication chain is invalid. ...

**(5)** A = { -7 < x < 0 }
B = { y = log(x) }
C = { -5 < x $\leq$ 10 }
D = { x = 1 $\wedge$ y $\geq$ 1 }
E = { -1 $\leq$ x $\leq$ 1 }

Implication Chain: $(E) \to (C) \to (D)$
Proof of implication: $A \to B$ : This is false because x being between -1 and 1 shows that it is also between -5 and 10, and contains the value of 1. y=(logx) wont work because if it is 1, then y must equal 0, but that statement is false. So the implication chain is invalid.

**(6)** `Assume "result is a double" and` $|\cdot|$ `is the absolute value operator.`
A = { `|result - sin(x)|` $\leq$ 1 }
B = { `|result - sin(x)|` $\leq$ 0.01 }
C = { `|result - sin(x)|` $\leq 10^{-10}$ }
D = { `|result - sin(x)|` $\leq$ -0.01 } = { false }
E = { `|result - sin(x)|` $\geq x^2$ }

Implication Chain: $(C) \to (B) \to (A)$
Proof of implication: $A \to B$ : This is True ...

**Problem 2 (4 pts., 1 pt. each): Hoare Triples**

State whether each Hoare triple is valid. If it is invalid, explain why and show how you would modify the the postcondition to make it valid. Unless otherwise stated, assume all referenced variables are defined as integers.

**(1)** { x = 5 }

```
x = x * 2;
```

{ x = 10 ∨ x ≠ 0}

Valid or Invalid: (Valid)

**(2)** { $\sqrt{x-1}$ > k }

```
x = x + 1;
```

{ k ≥ 0}

Valid or Invalid: (Invalid)

It's invalid because if the value of x is a negative number, then the square root of x-1 will be an imaginary number. To fix this, we can modify the postcondition to { k ≥ 0 ∧ x ≥ 1} as one valid solution.

**(3)** { i + j ≠ 0 ∧ i · j ≠ 0}

```
i = j - 1;
j = i + 1;
```

{ (i = 0 ∨ i ≠ -j) ∧ k ∈ ℚ }

Valid or Invalid: (Valid)

**(4)** { n < 0 ∧ n = $\sqrt{m}$ }

```
if (n > m)
  x = n;
else
  y = m;
```

{ x ≠ y }

Valid or Invalid: (Invalid)

It's invalid because you cant have a negative number as the square root of a number. To fix this, we can modify the precondition to { n ≥ 0 ∧ n = $\sqrt{m}$ } as one valid solution.

## Problem 3 (4 pts., 1 pt. each): General Hoare Triples

A, B, C, D, E, and F are logical conditions (logical formulas). The following are true:

- A $\rightarrow$ B (A implies B, i.e., A is stronger than B)
- B $\rightarrow$ C
- C $\rightarrow$ B
- D $\rightarrow$ E
- E $\rightarrow$ F
- { B } code { E }

Indicate whether the following Hoare triples are valid or possibly invalid. If possibly invalid, prove by giving an example.

**(1)** {C} `code` {D} Valid or Possibly Invalid: (Invalid)
One example of a possibly invalid case is if C is true and D is false. This would mean that C $\rightarrow$ B is true, but B $\rightarrow$ C is false. This would make the Hoare triple invalid.

**(2)** {B} `code` {C} Valid or Possibly Invalid: (Valid)
B implies C, so the Hoare triple is valid.

**(3)** {A} `code` {D} Valid or Possibly Invalid: (Invalid)
One example of a possibly invalid case is when A implies B, which implies C, but it never implies D.

**(4)** {A} `code` {F} Valid or Possibly Invalid: (Valid)
A implies B, which implies C, which implies B, which implies E, which implies F. So A implies F, making the Hoare triple valid.

## Problem 4 (8 pts., 1 pt. for each condition): Forward reasoning

For each code snippet with the given precondition, find the **strongest postcondition** by inserting the appropriate condition in each blank. The first intermediate condition in part (1) is supplied as an example. Please simplify your answers as much as possible. Assume all referenced variables are defined as integers.

Copy all code to your answer file and fill in the blanks. Carry all variables forward. Show all work.

**(1)**
```
{ z ≠ 0 }
y = 0;
    { y = 0 ∧ z ≠ 0 }
x = y + 2;
   { x = 2 ∧ y = 0 ∧ z ≠ 0 }
z = x + y;
{ x = 2 ∧ y = 0 ∧ z = 2 }
```

**(2)** { |x| > 5 }
```
x = x % 10;
```
{ 0 ≤ |x| ≤ 9 }
```
x = x * x;
```
{ 0 ≤ |x| ≤ 81 }
```
x = -x;
```
{ -81 ≤ |x| ≤ 0 }

**(3)** { z < 5 }
```
if (z >= 0) {
```
{ 0 ≤ z < 5 }
```
z = -z;
```
{ -5 < z ≤ 0 }
```
}
```
{ z ≤ 0 }

## Problem 5 (11 pts., 0.5 pts. each condition): Backward reasoning

Find the **weakest precondition** of each code sequence by inserting the appropriate condition in each blank. The first intermediate condition in part (1) is supplied as an example. Please simplify your answers as much as possible. Assume all referenced variables are defined as integers. Use the `wp()` notation as shown in the first example.

**(1)** { `wp("x = -1", y > -3x ∧ y < 10 - 3x )` = y > 3 ∧ y < 13 }

x = -1;

{ wp("z = 2 * y + x", 0 < z < 10) } = { 0 < 2y + x < 10 } =

{ x > -2y ∧ x < 10 - 2y }

z = 3 * x + y;

{ 0 < z < 10 }

**(2)** `wp("if (y > 0)", (x > y ∧ y > 0) ∧ (False))` = x > y ∧ y > 0 }

if (y > 0) {

{ wp("x = x / y", x > 1 ∧ y = 0) = x > y ∧ y >0 }

x = x / y;

{ wp("y = 0", x > 1 ∧ y = 0) = x > 1 ∧ y = 0 }

y = 0;

{ ∧ x > 1 }

} else {

{ wp("y = 4 * x", x > 1 ∧ y = 0) = Statement isn't possible so else statement }

y = 4 * x;

{ y = 0 ∧ x > 1 }

}

{ y = 0 ∧ x > 1 }

**(3)** { `wp("if (x ≥ 0)", min(z,x) ≠ 0 ∧ y ≥ 0 ∨ min(z,x) ≥ -y)` = min(z,x) ≠ 0 ∧ y ≥ 0 ∨ min(z,x) ≥ -y }
if (x ≥ 0) {
{ wp("z = Math.min(z,x)", z ≠ 0 ∧ y ≥ 0 ∨ z ≥ -y) = min(z,x) ≠ 0 ∧ y ≥ 0 ∨ min(z,x) ≥ -y }
z = Math.min(z, x);
{ wp("x = z + y", z ≠ 0 ∧ y ≥ 0 ∨ x ≥ 0) = z ≠ 0 ∧ y ≠ 0 ∨ z ≥ -y }
x = z + y;
{ z ≠ 0 ∧ y ≥ 0 ∨ x ≥ 0 }
}
{ z ≠ 0 ∧ y ≥ 0 ∨ x ≥ 0}

**(4)** { `wp("if (Math.abs(x) ≤ 5)", (-1 ≤ x ≤ 5) ∨ (-9 ≤ x ≤ 1))` = -9 ≤ x ≤ 5 if (Math.abs(x) ≤ 5) }
if (Math.abs(x) <= 5) {
{ wp("z = x-2", -3 ≤ z ≤ 3) = -1 ≤ x ≤ 5 z = x - 2; }
z = x - 2;
{ -3 ≤ z ≤ 3 }
} else {
{ wp("if (x ≤ -5)", (-9 ≤ x ≤ -3) ∨ (-1 ≤ x ≤ 1)) = -9 ≤ x ≤ 1 }
if (x <= -5) {
{ wp("z = x + 6", -3 ≤ z ≤ 3) = -9 ≤ x ≤ -3 }
z = x + 6;
{ -3 ≤ z ≤ 3 }
} else {
{ wp("z = 3 * x", -3 ≤ z ≤ 3) = -1 ≤ x ≤ 1 }
z = 3 * x;
{ -3 ≤ z ≤ 3 }
}
{ -3 ≤ z ≤ 3 }
}
{ -3 ≤ z ≤ 3 }

**(5)** { `wp("x = y / 2", x ≠ 0.5 ∧ x > -1)` = y ≠ 1 ∧ y > -2 }
x = y / 2;
{ wp("z = x + 1", x ≠ 0.5 ∧ z > 0) = x ≠ 0.5 ∧ x > -1 }
z = x + 1;
{ x ≠ 0.5 ∧ z > 0}

**Problem 6 (10 pts., 1 pt. each condition, 1 pt. sufficient/insufficient): Verifying Correctness**

For each block of code, fill in all the conditions, then use them to state whether the precondition is sufficient to guarantee the postcondition. If the precondition is insufficient, explain why.

Hint: Use backward reasoning to find the weakest precondition that guarantees the postcondition and see if the given precondition is strong enough to guarantee the postcondition. In other words, is the given precondition not stronger than the weakest precondition?

Copy all code to your answer file and fill in the blanks. Answers must be expressed in the format `wp("code",␣condition)␣=␣precondition`. Show all work. Assume all referenced variables are defined as integers.

**(1)** { x < 2 }

```
    { x < 2 }
z = x - z;
    { wp("w = x - 1", w > 2) }
w = x - 1;
    { wp("z = w - 1", z > 1) }
z = w - 1;
```

{z > 1}

Sufficient or Insufficient: { Insufficient because our precondition won't guarantee that our postcondition is true, but to guarantee it, use x > 3 }

**(2)** { x = y ∧ y

>

0 ∨ y ≠ x ∧ x ≥ 0 }

{ wp("if (x < y)", (x ≥ y ∧ x < y ∧ y > 0) ∨ (x < y ∧ x ≤ y ∧ x ≥ 1)) = x = y ∧ y > 0 ∨ x < y ∧ x ≥ 1 }

if (x < y) {

{ wp("x- -", x < y ∧ x ≥ 0) = x ≤ y ∧ x ≥ 1 }

x–;

{ x < y ∧ x ≥ 0 }

} else {

{ wp("x = y / 2", x < y ∧ x ≥ 0) = x < y ∧ y > 0 }

x = y / 2;

{ x < y ∧ x ≥ 0 }

}

{ x < y ∧ x ≥ 0}

Sufficient or Insufficient: { Insufficient because our precondition won't guarantee that our postcondition is true, but to guarantee it, use x = y }